

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інтегровані інформаційні
системи»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Вебзастосунок для замовлення квитків на футбольні
матчі»**

Виконав:

студент IV курсу, групи ІА-93

Дюг Андрій Ігорович _____

Керівник:

Доцент Крилов Є.В. _____

Рецензент:

Доцент Лісовиченко О.І. _____

Засвідчую, що у цьому
дипломному проєкті немає
запозичень з праць інших авторів
без відповідних посилань.
Студент _____

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Дюгу Андрію Ігоровичу

1. Тема проєкту «Вебзастосунок для замовлення квитків на футбольні матчі», керівник проєкту доцент Крилов Є.В., затверджені наказом по університету від «31» травня 2023 р. №2101-с
2. Термін подання студентом проєкту
12 червня 2023 р.
3. Вихідні дані до проєкту
Мова гіпертекстової розмітки HTML, каскадна таблиця стилів CSS, мова програмування JavaScript, мова програмування PHP, база даних MySQL, програмний інструмент phpMyAdmin, середовище програмування Visual Studio Code.
4. Зміст пояснювальної записки
Дослідити доцільність розроблюваною системи, проаналізувати перелік технологій для розробки застосунку, порівняти систему із уже розробленими рішеннями, зробити порівняльну характеристику розробленої системи з готовими рішеннями.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

Діаграма послідовності, діаграма розгортання, діаграма діяльності, діаграма компонентів.

б. Дата видачі завдання

1 березня 2023р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Узгодження теми роботи	28.02.2023	
2.	Аналіз предметної області	15.04.2023	
3.	Розробка завдання, огляд існуючих рішень	30.04.2023	
4.	Проектування системи	05.05.2023	
5.	Розробка проєкту	17.05.2023	
6.	Тестування програми	19.05.2023	
7.	Оформлення пояснювальної записки	02.06.2023	
8.	Передзахист дипломного проєкту	04.06.2023	
9.	Доопрацювання пояснювальної записки	12.06.2023	
10.	Підготовка презентації	14.06.2023	
11.	Захист дипломної роботи	19.06.2023	

Студент
Керівник

Андрій Дюг
Євген Крилов

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з п'яти розділів, містить 26 рисунків, 2 таблиці, 1 додаток, 19 джерел – загалом 72 сторінки.

Дипломний проект присвячений розробці застосунку для віддаленого замовлення квитків на футбольні матчі. Метою роботи є скорочення часу очікування квитка та мінімізування черг під час проходження квиткового контролю шляхом модернізації механізму роботи квитка.

У розділі огляд існуючих рішень було описано існуючі застосунки зі схожою функціональністю та представлено їх переваги і недоліки.

У розділі вибір підходів та технологій для розробки описано основні засоби розробки застосунку і визначено вимоги до технічного забезпечення.

Розробка застосунку описує архітектуру та реалізацію програмного забезпечення.

У розділі тестування програмного забезпечення проведено випробування програмного продукту.

ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of five sections, contains 26 figures, 2 tables, 1 appendix, 19 sources – 72 pages in total.

The diploma project is devoted to the development of an application for remote ordering of tickets for football matches. The goal of the work is to reduce the waiting time for a ticket and minimize queues during ticket control by modernizing the ticket mechanism.

In the overview of existing solutions section, existing applications with similar functionality were described and their advantages and disadvantages were presented.

In the section on the selection of approaches and technologies for development, the main means of application development are described and the requirements for technical support are defined.

Application development describes the architecture and implementation of software.

In the software testing section, the software product is tested.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			Документація загальна			
2						
3			Знову розроблена			
4						
5	A4	IA93.110БАК.005 ПЗ	Пояснювальна записка	61		
6	A3	IA93.110БАК.005 Д1	Діаграма діяльності	1		
7						
8	A3	IA93.110БАК.005 Д2	Діаграма розгортання	1		
9						
10	A3	IA93.110БАК.005 Д3	Діаграма послідовності	1		
11						
12	A3	IA93.110БАК.005 Д4	Діаграма компонентів	1		
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
				IA93.110БАК.005 ТП		
Зм.	Аркуш	№ докум.	Підпис	Дата		
Розроб.		Дюг А.І.			Літ.	Аркуш
Керівн.		Крилов Є.В.			Т	Аркушів
						1
						1
Затв.					КПІ ім. Ігоря Сікорського Група ІА-93	
Вебзастосунок для замовлення квитків на футбольні матчі. Відомість проекту						

**Пояснювальна записка
до дипломного проекту
на тему: «Вебзастосунок для замовлення квитків
на футбольні матчі»**

Київ – 2023 року

ЗМІСТ

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ	4
ВСТУП	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Функціональні вимоги до проєкту	9
1.2 Нефункціональні вимоги до проєкту	14
1.3 Постановка задачі	15
1.3.1 Призначення розробки	21
1.3.2 Цілі та задачі розробки.....	15
2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	17
2.1 TicketsBox	17
2.2 eSport	19
2.3 Kontramarka.....	20
3 ВИБІР ПІДХОДІВ ТА ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ	23
3.1 Технологічний стек.....	23
3.2 Мова гіпертекстової розмітки HTML	23
3.3 Каскадна таблиця стилів CSS	25
3.4 Мова програмування JS.....	26
3.5 Мова програмування PHP	27
3.6 База даних MySQL.....	30
3.7 SQL-запити	31
4 РОЗРОБКА ЗАСТОСУНКУ	33
4.1 Архітектура сервісу	33
4.2 Розробка веб-застосунку	35
4.3.1 Модуль системи керування переліком матчів	36
4.3.2 Модуль онлайн замовлень	37
4.3.3 Модуль обробки замовлення	39

					ІА93.110БАК.005 ПЗ			
Зм.	Лист	№ докум.	Підпис		Вебзастосунок для замовлення квитків на футбольні матчі. Пояснювальна записка	Літ.	Арк.	Аркушів
Розробив	Дюг А.І..					Т	2	65
Перевірив	Крилов Є.В.				КПІ ім. Ігоря Сікорського Група ІА-93			
Затв.								

4.4	Діаграма послідовності.....	41
4.5	Діаграма розгортання	42
4.6	Діаграма діяльності.....	42
4.7	Діаграма компонентів.....	42
5	ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	44
5.1	Мета тестування.....	44
5.2	Загальні положення.....	44
5.3	Функціональне тестування.....	44
5.4	Інтеграційне тестування	50
6	ПОРІВНЯННЯ ПРОЄКТУ ТА ГОТОВИХ РІШЕНЬ	55
	ВИСНОВКИ.....	58
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
	ДОДАТОК А.....	62

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ

API — Application Programming Interface — сукупність методів взаємодії однієї комп'ютерної програми з другими для створення програмного забезпечення

HTML – Hyper Text Markup Language – стандартизована мова розмітки документів, яка призначена для перегляду веб-сторінок у браузері

CSS – Cascading Style Sheets – мова стилів, яку використовують для опису вигляду документу

JS — JavaScript — об'єктно-орієнтована мова програмування

БД – база даних

CRUD – Create, Read, Update, Delete – основні функції керування даними: створення, читання, оновлення, видалення

Користувач – будь-яка людина, що користується застосунком

СУБД — система управління базами даних

ПЗ – програмне забезпечення

MVC — Model-View-Controller — шаблон проєктування та розробки програмного забезпечення, який передбачає поділ системи на три частини: модель, вигляд (інтерфейс), модуль керування

ПК – персональний комп'ютер

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

ВСТУП

Ми живемо в ХХІ столітті – періоді новітніх технологій. В цей час розробники програмного забезпечення зуміли розробити велику кількість програмних засобів, які полегшують не тільки робочий процес, але й дозвілля. Сотні програм дозволяють людям економити їх дорогоцінний час на спілкування з сім'єю, або друзями, замість роботи за комп'ютером. До прикладу, штучний інтелект може зробити роботу, на яку людина витратила б дві години, за декілька секунд.

Але у цей період глобальних розробок люди зовсім не звертають уваги на створення власного комфорту. Для замовлення квитка на будь-яке спортивне змагання або похід в парк атракціонів людина повинна звернутись у касу, провести декілька годин свого життя в очікуванні своєї черги, а потім стільки ж у черзі біля стадіону чи атракціону. Звичайно, замовити квиток можна і онлайн, проте у більшості випадків цей процес є настільки складним та заплутаним, що більшість людей обирає варіант замовлення квитка в касі. Про яку економію часу тоді взагалі може іти мова? На мою думку, в наш час цей процес повинен піддатись негайній цифровізації та оптимізації.

Футбольні змагання є популярними не тільки в країнах Європи, але й у всьому світі. Щодня ведуться десятки матчів різних ліг та між різними командами. Щодня за даними подіями спостерігають тисячі, а то й мільйони глядачів онлайн. Проте найвідданіші фанати команд ідуть далі – вони купують квиток на матч своєї улюбленої команди, щоб відчувати цю атмосферу стадіону, наживо побачити матч, а також своїх кумирів.

Було б непогано, якби існував такий сервіс, який би спеціалізувався тільки на продажі квитків на футбол, але б використовував сучасні розробки, був би із зрозумілим інтерфейсом та передбачуваним у користуванні. Це дозволило б зекономити вболівальникам час на замовленні квитків, а обслуговуючому персоналу – на перевірку квитків та допуску людей до перегляду матчу.

					ІА93.110БАК.005 ПЗ	Арк.
						5
Зм.	Лист	№ докум.	Підпис	Дата		

У пояснювальній записці буде зібрано опис основних технологій, які використовуються для розробки застосунку, деталі реалізації та управління користувача.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Кожного дня у світі з'являється все більше і більше веб-сайтів. Це легко пояснити тим, що у теперішній час веб-застосунки з легкістю витісняють програми, які в більшості випадках є застарілими. Веб-застосунки є не тільки зручними для користувача, але й для розробника, оскільки як правило, версії для персональних комп'ютерів та мобільних телефонів одні і ті ж, тільки дещо видозмінені.

Звичайно, веб-сайти не можуть замінити потужні програми, тому говорити про тотальну монополію веб-застосунків не можна, але легкі додатки, наприклад інтернет-магазин, замінити можна з легкістю. В більшості випадків так і є, тому розробники розробляють і веб-застосунок, і десктопну програму, щоб задовольнити інтереси більшості користувачів і цим же привернути більше аудиторії до свого продукту.

У зв'язку з постійним розвитком технологій, кількість веб-сайтів, які підтримують адаптивність стрімко зростає. Адаптивність додають навіть до тих застосунків, які не потребують цього. Це пояснюється тим, що у світі постійно збільшується кількість смартфонів, а саме адаптивність дозволяє коректно відображати веб-сайт як на ПК, так і на смартфоні.

На рисунку 1.1 приведено статистику росту кількості смартфонів у світі у період з 2014 року до 2020 року.

Як видно з діаграми, у 2020 році, кількість людей, що мали смартфон сягала 3 млрд. осіб. Проте у 2021 році ситуація змінилась ще виразніше – кількість користувачів смартфонів зросла до 4 млрд. осіб. Винуватцем такого росту кількості смартфонів стала коронавірусна хвороба, яка відправила на дистанційну роботу практично всіх людей. Зараз кількість смартфонів у світі перевищує 5 млрд. шт. Звичайно, їх ріст уже не такий, як у 2021 році, але він все ж присутній.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		7

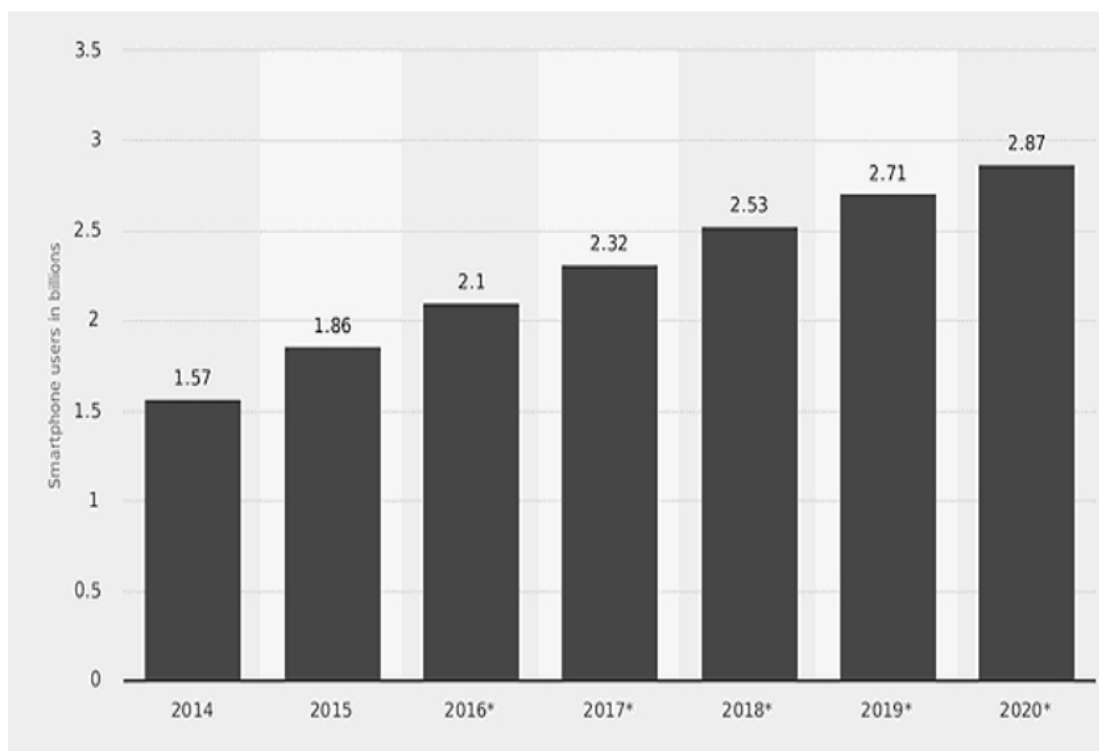


Рисунок 1.1 – Діаграма росту кількості смартфонів у світі [1]

Більшість замовників зараз включає адаптивність веб-застосунку в їх список опцій. Зараз замовник хоче зробити так, щоб його продукт був доступним користувачеві в будь-якій точці світу і в будь-якій ситуації.

Говорячи про переваги веб-застосунків над стільниковими програмами можна виділити кілька основних:

- вони не мають прив'язки до типу пристрою та операційної системи;
- на відміну від деяких програм, інформація, розміщена на них є безкоштовною;
- їх не потрібно завантажувати на пристрій;
- браузер знаходить сотні сторінок веб-сайтів, в яких міститься запитувана інформація, і лише кілька додатків, у яких вона може міститись;
- простота підтримки застосунку для розробника

Проте є у веб-застосунків і мінуси, і найбільший із них – це необхідність наявності доступу до мережі Інтернет. Більшість програм зберігають переглянуту інформацію у пам'ять пристрою, тому вона може бути переглянута і без

подальшого підключення до мережі Інтернет, чим не може похвалитись веб-застосунок, оскільки останній опрацьовується браузером.

1.1 Функціональні вимоги до проєкту

Функціональні вимоги описують поведінку системи в різних сценаріях роботи. Вони ставлять завдання розробникові, які питання повинна вирішувати розроблювана ним система.

Основною функціональною вимогою даного застосунку є надання користувачеві можливості замовити квиток на футбольний матч. Застосунок повинен працювати на пристроях на основі таких операційних систем:

- Windows
- Linux
- MacOS
- Android
- ISO

Окрім замовлення квитка на футбольний матч, важливою функціональною вимогою до проєкту є надання користувачеві можливості ознайомитись із доступними місцями на трибуні стадіону, та ознайомитись із вартістю квитка на кожне із них.

Також повинно бути розроблено систему облікового запису. Користувач повинен мати змогу зареєструватись в веб-застосунку, увійти в уже створений аккаунт, а також вийти зі свого облікового запису. Окрім того, повинна проводитись перевірка на наявність облікового запису у списку створених. У випадку, якщо у розділі реєстрації користувач вводить дані, які відповідають тим, що знаходяться у списку вже створених, то користувача має допускати до його облікового запису. У випадку, якщо дані відрізняються, повинен бути зареєстрований ще один обліковий запис. Якщо користувач проводить авторизацію свого облікового запису і дані не збігаються з тими, що були надані під час реєстрації, йому має сповіщатись про невідповідність даних.

					ІА93.110БАК.005 ПЗ	Арк.
						9
Зм.	Лист	№ докум.	Підпис	Дата		

Якщо користувач зареєструвався в веб-застосунку або увійшов в обліковий запис за допомогою авторизації, то користувачеві повинна надаватись можливість забронювати місце на трибуні стадіону та перегляд сформованого квитка у розділі «Корзина».

Окрім користувача потрібно розробити зручний інтерфейс і для адміністратора. Отже необхідно розробити сторінку адміністратора, доступ до якого буде відбуватись після введення унікального логіну та паролю. Сторінка адміністратора повинна надавати адміністратору можливість створення нових квитків та матчів, редагування вже існуючих, або видалення матчів з переліку. Тобто, дана сторінка повинна реалізовувати CRUD операції.

Дані користувачів зберігаються в базі даних. Окрім того, в базі даних зберігається і перелік замовлень квитків, а також інформація про замовника. Авторизований обліковий запис надає користувачеві доступ до повного функціоналу веб-сайту.

Аналізуючи вищенаведений матеріал, можна заявити, що акторами, які будуть використовувати даний веб-застосунок, є:

- Адміністратор
- Незареєстрований користувач
- Зареєстрований користувач

Поділ користувача на незареєстрованого та зареєстрованого пояснюється тим, що функціонал, який буде доступний кожному із них відрізняється.

На рисунку 1.2 наведено основні функції системи, та авторів, що їх виконують.

					ІА93.110БАК.005 ПЗ	Арк.
						10
Зм.	Лист	№ докум.	Підпис	Дата		

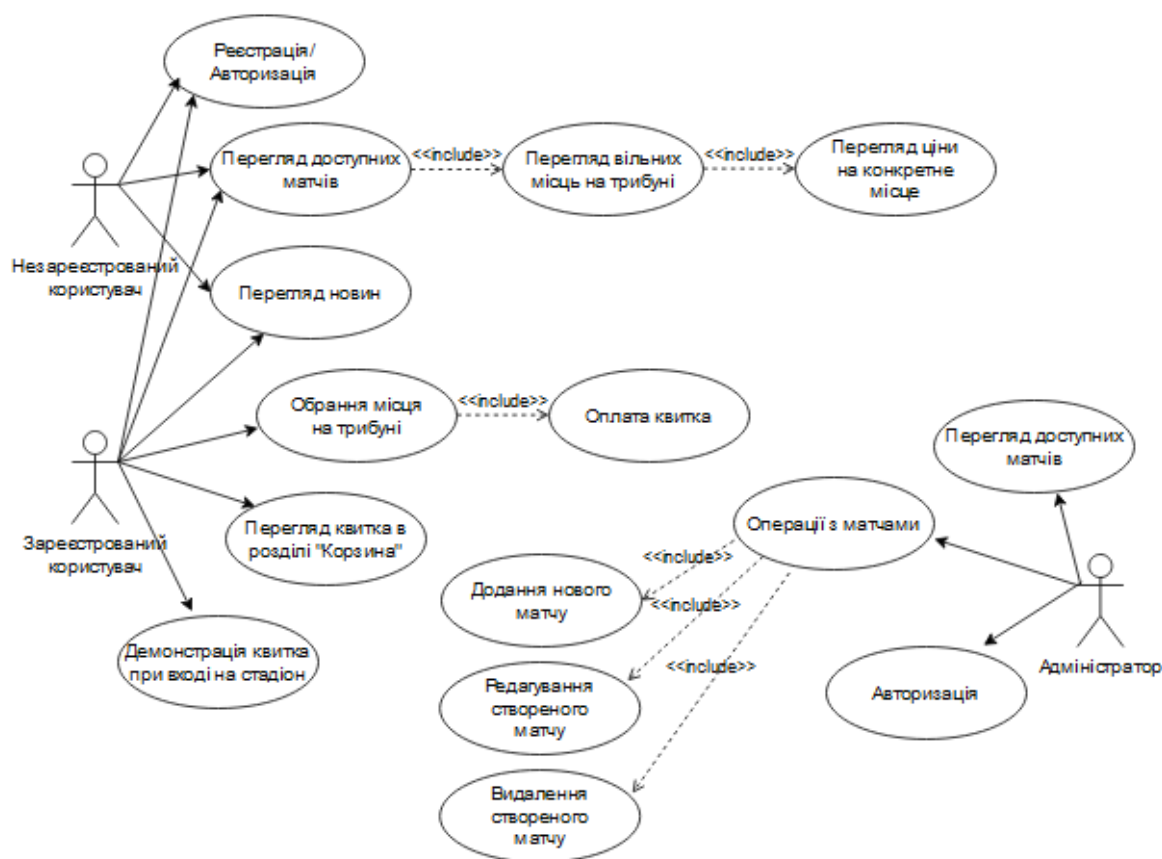


Рисунок 1.2 – Use-case діаграма

Опис функціональних вимог для актора «Незареєстрований користувач» наведений в таблиці 1.1

Таблиця 1.1 – Перелік функціональних вимог для актора «Незареєстрований користувач»

Операція	Опис функціональної вимоги
Реєстрація	Для створення облікового запису користувачу потрібно натиснути на відповідну кнопку в навігаційному меню, яка перенаправляє користувача на відповідну сторінку. У відповідні поля ввести дані користувача. У випадку успішної реєстрації, користувача буде перенаправлено на головну сторінку. В іншому випадку, користувачеві буде виведено помилку.

Авторизація	Для проведення процесу авторизації користувач має натиснути відповідну кнопку в навігаційному меню (ту ж, що й для проведення процесу реєстрації) та ввести необхідні дані у відповідні поля. У випадку успішної авторизації користувача перенаправить на головну сторінку веб-сайту. В іншому випадку, користувач отримає помилку входу.
Перегляд доступних матчів	Для перегляду доступних матчів користувачеві достатньо потрапити на головну сторінку веб-сайту. На ній зібрано всі доступні матчі команд різних ліг: ліги України, ліги Європи, та прем'єр ліги. Крім того, перелік доступних матчів зібрано у вкладці «Квитки». Крім того, у розділі «Квитки» наведено дату проведення матчу та кількість вільних місць на трибуні стадіону.
Перегляд вільних місць на трибуні	Для того, щоб користувач міг побачити наочно, які місця на трибуні стадіону залишились вільними для бронювання, йому достатньо обрати матч, який його цікавить. Після цього застосунок перенаправить його на сторінку з інтерактивною мапою стадіону. По контуру стадіону розміщені кнопки, які і є місцями на трибуні. Замальовані кнопки – це заброньовані місця. Відповідно, всі інші кнопки – це місця, доступні для бронювання.
Перегляд ціни на місце	Для того, щоб користувач міг побачити ціну на квиток за місце на трибуні, йому потрібно виконати крок «Перегляд вільних місць на трибуні», після цього натиснути на місце, яке

	цікавить користувача, за умови, що воно не заброньоване. На бічній панелі (для користувачів ПК) або внизу сторінки(для користувачів смартфонів) буде виведено інформацію про номер місця, то його ціну. При множинному виборі місць ця інформація теж буде виводитись, а також буде виводитись загальна сума чеку.
Перегляд новин	Для того, щоб користувач міг почитати новини футболу, йому достатньо натиснути вкладку «Новини» в навігаційному меню. Веб-сайт перенаправить користувача на сторінку з новинами.

Оскільки після проходження етапу реєстрації або авторизації незареєстрований користувач стає зареєстрованим, то і список функціональних вимог для нього розширюється. В таблиці 1.2 наведено перелік функціональних вимог, які додаються користувачеві після проходження етапу реєстрації/авторизації.

Таблиця 1.2 – Перелік функціональних вимог для актора «Зареєстрований користувач»

Оплата квитка	Після того, як користувач ознайомився з ціною на місця на трибуні, які його цікавлять, він може перейти до етапу оплати квитка, натиснувши кнопку «Замовити». Веб-сайт виведе користувачеві форму, яку він повинен заповнити для проведення оплати, та натиснути кнопку «Придбати». Якщо всі поля введені вірно, користувача буде переадресовано на головну сторінку веб-сайту. В іншому випадку,
---------------	---

	користувачеві буде виведено помилку про необхідність надання вірних даних.
Перегляд квитка в розділі «Корзина»	Після успішної оплати за квиток на матч користувач може переглянути його, натиснувши відповідну кнопку в навігаційному меню веб-сайту. Веб-сайт переадресує користувача на сторінку покупок, де у формі QR-коду буде відображено квиток на футбольний матч. На бічній панелі буде виведено місця на трибуні, які обрав користувач. Окрім того, відсканувавши QR-код, він отримає інформацію про те, на який матч користувач замовив квиток, та місця на трибуні, які він обрав.

1.2 Нефункціональні вимоги до проекту

Нефункціональні вимоги описують вимоги до системи в цілому, описують характеристики поведінки системи в певних випадках та обмеження, яких повинна дотримуватись система.

Основними нефункціональними вимогами, які виділені до цього проекту є:

- адаптивність – веб-сайт повинен правильно та зручно відкриватись на будь-якому пристрої;
- мультиплатформність – застосунок повинен правильно працювати на будь-якій операційній системі та в будь-якому браузері;
- легкість – застосунок повинен бути інтуїтивно зрозумілим для користувача різної вікової категорії;
- продуктивність – застосунок повинен виконувати операції швидко, без довгих очікувань.

1.3 Постановка задачі

1.3.1 Призначення розробки

Запропонований веб-застосунок призначений для придбання квитків на футбольні матчі. Його завданням є модернізація процесу замовлення квитків та спрощення процесу проходження квиткового контролю шляхом модернізації технічної складової застосунку. Застосунок мінімізує час очікування користувача на різних етапах замовлення квитка.

1.3.2 Цілі та задачі розробки

Мета роботи – зекономити час проведення квиткового контролю шляхом модернізації квитків.

Веб-застосунок повинен дозволяти користувачам проводити бронювання місця на трибуні стадіону, правильно відображати інформацію про доступні та недоступні місця, а також правильно відображати інформацію для користувачів бідь-якого пристрою – від ПК до смартфона. Окрім того, квитки повинні зберігатись у відповідному розділі, для того, щоб користувач міг продемонструвати його під час проходження квиткового контролю. Адміністратори повинні мати змогу редагувати інформацію про матчі, додавати нові матчі та видаляти існуючі.

Для реалізації поставлених цілей необхідно:

1. Створити на налаштувати зв'язок з базою даних
2. Створити клієнтську частину для простоти роботи із застосунком
3. Створити серверну частину для опрацювання даних і подальшого відправлення їх на клієнтську частину або базу даних
4. Налаштувати можливість створення облікових записів, реалізації реєстрації на сайті та авторизації
5. Створити можливість проведення оплати за квиток за допомогою застосунку
6. Описати медіа запити для коректного відображення застосунку на різних пристроях.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		15

Висновки до розділу: в цьому розділі було проведено функціональні та нефункціональні вимоги до проєкту, а також було наведено їх детальний опис. Було побудовано діаграму функціональних вимог, складено таблицю функціональних вимог для акторів «незарєєстрований користувач» та «зарєєстрований користувач». Було визначено призначення розробки а також було оформлено цілі та задачі для розробки проєкту.

					IA93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		16

2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

З плином часу люди дедалі більше розвивають технології задля економії свого часу з метою використання його на свої потреби, а не на роботу. У світі з'являється все більше і більше систем, які допомагають людям працювати у сфері надання різних послуг. В тому числі, це стосується і сервісів для продажу квитків на футбольні матчі.

Серед найбільш відомих сервісів, які надають послуги продажу квитків на футбольні матчі в Україні є:

- TicketsBox [2]
- eSport [3]
- Kontramarka [4]

Кожен із них є схожим за сферою надання послуг, але одночасно різний починаючи від дизайну, закінчуючи стадіонами, з якими працюють сервіси. Розглянемо кожен із них детальніше.

2.1 TicketsBox

TicketsBox – сервіс, що дозволяє придбати квитки не тільки на футбольні матчі, а й на вистави, концерти, та інше. Зосереджений на роботі в дев'ятнадцяти містах України, в тому числі на території Кримського півострова.

Перевагами застосунку є:

- зручна навігація та наявність фільтрів
- підтримка роботи з великою кількістю міст України
- робота на будь-яких пристроях
- приємний дизайн
- можливість надання інформації не тільки українською, а й російською та англійською мовами.

Недоліки застосунку:

- відсутність роботи зі стадіонами України, тобто квитки на матчі продаються через сторонні сервіси

					ІА93.110БАК.005 ПЗ	Арк.
						17
Зм.	Лист	№ докум.	Підпис	Дата		

- відсутність можливості збереження квитка прямо на сайті – його потрібно зберігати на пристрій або роздруковувати
- повільна робота сайту.

На рисунку 2.1 зображено інтерфейс системи для замовлення квитків в Україні TicketsBox.

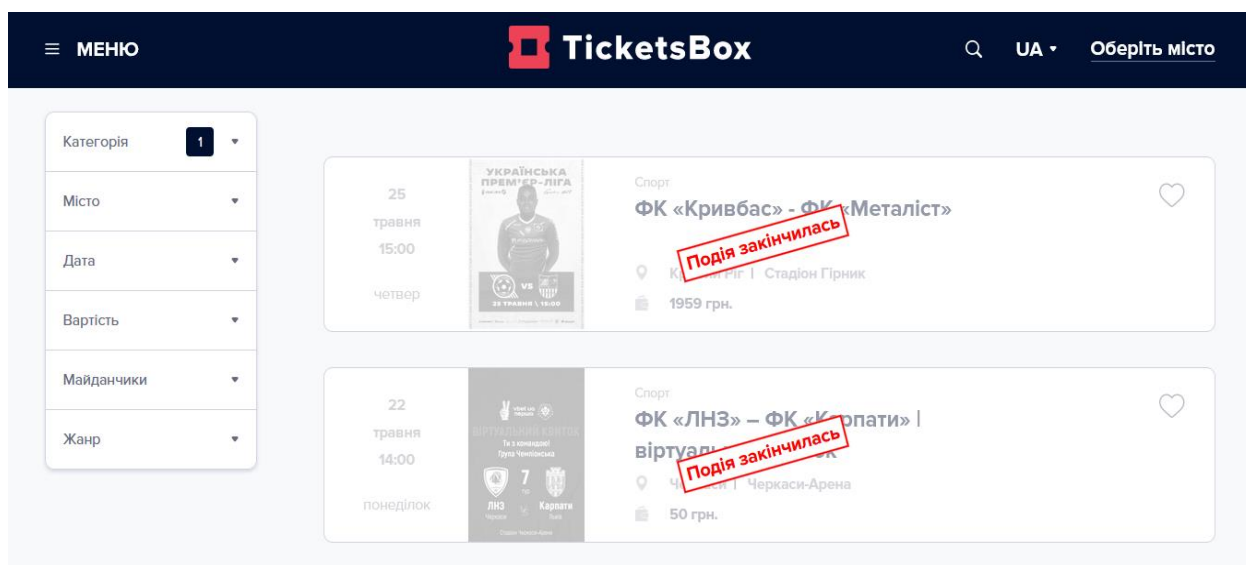


Рисунок 2.1 – Інтерфейс системи TicketsBox

Перелік міст, з якими працює сервіс, наведений на рис 2.2.

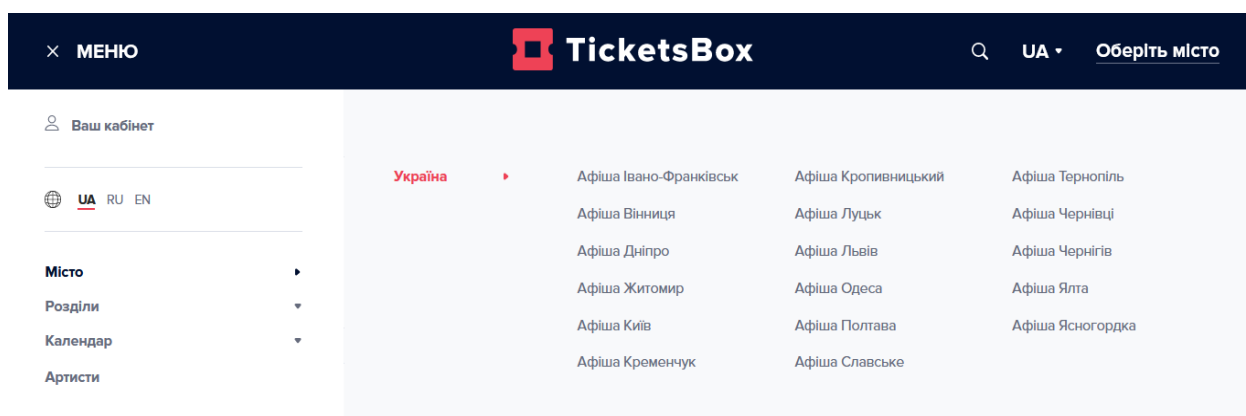


Рисунок 2.2 – Перелік підтримуваних міст

2.2 eSport

eSport – сервіс, який надає послуги бронювання місць на спортивні змагання, які проводяться на території України. Серед видів спорту, які підтримує сервіс знаходяться:

- футбол
- хокей
- баскетбол
- джиу-джитсу

Віднедавна сервіс почав підтримувати продаж квитків на концерти та в кінотеатри України. Сервіс працює із двадцятьма п'ятьма містами України а також з трьома стадіонами, такими як:

- НСК «Олімпійський»
- Стадіон «Динамо»
- Арена «Львів»

Перевагами сервісу є:

- офіційна підтримка великих стадіонів України, тому продаж квитків відбуватиметься безпосередньо з кас цих стадіонів

- наявність великої кількості спортивних подій
- наявність можливості надання інформації українською, російською та англійською мовами

Недоліки сервісу:

- розбіжності в назві та спеціалізації, оскільки на сайті, який був створений тільки для спортивних подій, з'явилися квитки на концерти та кіно
- застарілий дизайн та складне управління
- відсутність можливості збереження електронної версії квитка – для отримання квитка потрібно замовити та оплатити доставку або самостійно забрати квиток в касі стадіону.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		19

На рисунку 2.3 зображено інтерфейс системи eSport.

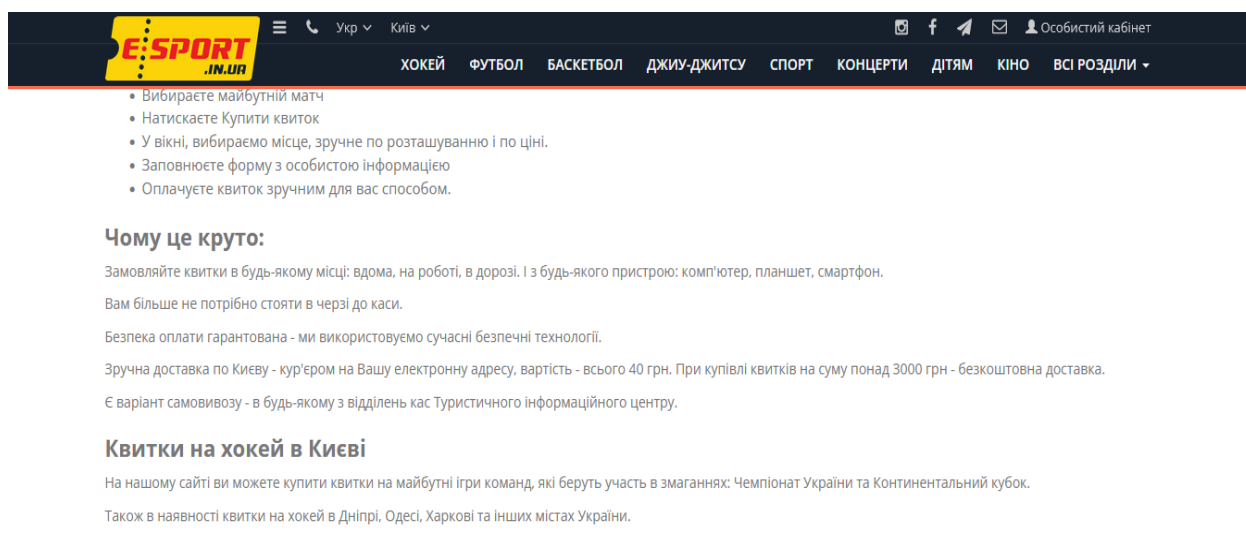


Рисунок 2.3 – Інтерфейс системи eSport

Перелік міст, з якими працює сервіс, зображений на рисунку 2.4

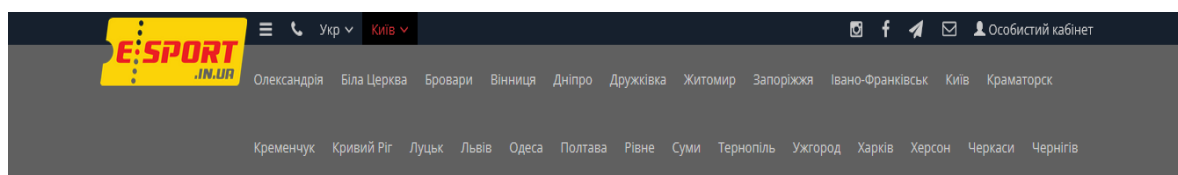


Рисунок 2.4 – Перелік підтримуваних міст

2.3 Kontramarka

Kontramarka – сервіс, який надає послуги продажу квитків на концерти та спортивні події. Окрім Українських подій, працює ще з шістнадцятьма країнами Європи. В Україні працює з п'ятдесятьма чотирма містами та містечками, що значно виділяє сервіс з-поміж інших розглянутих. Окрім веб-сайту, має програмний застосунок, який можна завантажити з Google Play або App Store. На території України має чотири каси, три з яких розташовані у місті Києві, та одна у Львові.

Переваги сервісу:

– підтримка подій у великій кількості міст України та Європи

					ІА93.110БАК.005 ПЗ	Арк.
						20
Зм.	Лист	№ докум.	Підпис	Дата		

- наявність великої кількості жанрів
- наявність програми для мобільних пристроїв
- наявність розділу бізнес подій
- наявність можливості надання інформації українською, російською та англійською мовами

- наявність меню пошуку бажаної події
- наявність гарячої лінії

Недоліки сервісу:

- розбіжності в спеціалізації, оскільки серед розважальних подій та концертів розміщені спортивні події
- наявність спливаючої реклами
- складний для розуміння інтерфейс
- спотворення виведення інформації під час перегляду на мобільних пристроях

- наявність банерів, які затуляють основний зміст
- відсутність можливості зберегти квиток в сервісі

На рисунку 2.5 наведено інтерфейс системи Kontramarka.

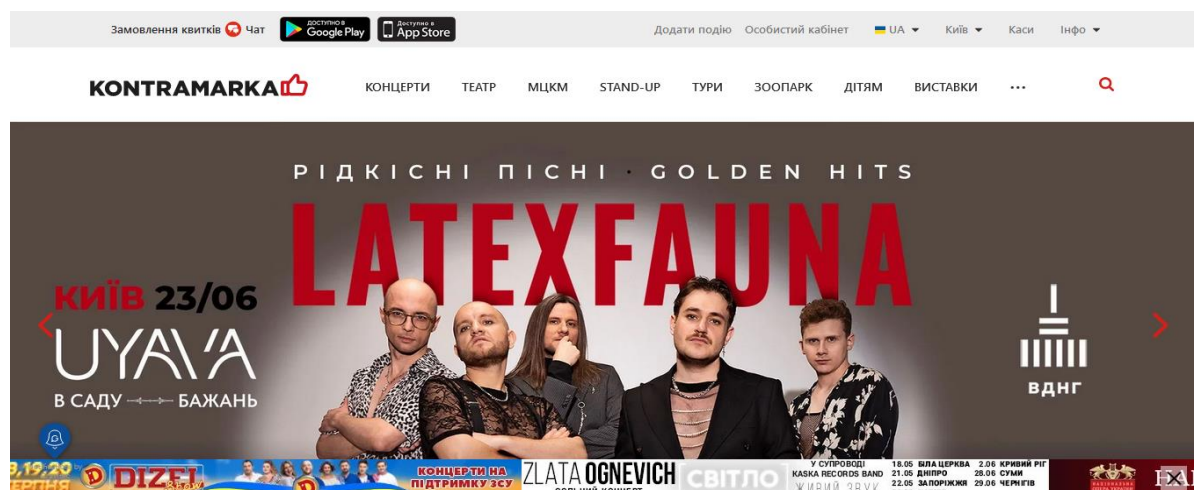


Рисунок 2.5 – Інтерфейс системи Kontramarka

Як видно із рисунку 2.5, на сайті присутні такі недоліки, як:

					IA93.110BAK.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		21

- наявність банерів, які затуляють основний зміст (нижче навігаційного меню)
- наявність спливаючої реклами (внизу рисунку)

На рисунку 2.6 наведено інтерфейс основної частини системи

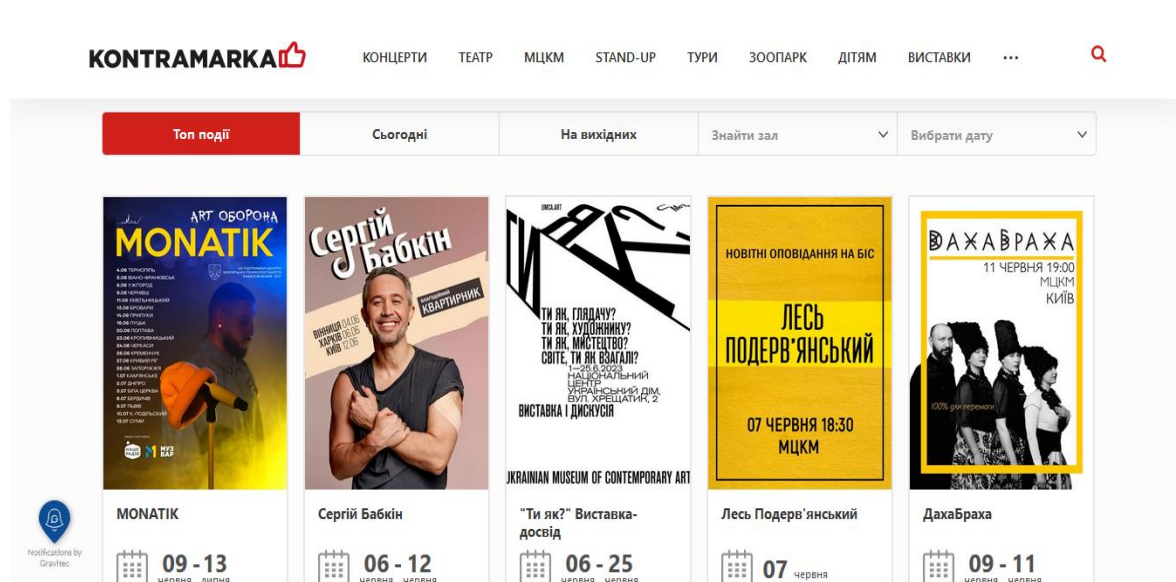


Рисунок 2.6 – Основний інтерфейс системи Kontramarka

Висновок до розділу: в цьому розділі я проаналізував існуючі рішення, які відповідають темі «Замовлення квитків на футбольні матчі». В результаті дослідження можу сказати, що жодна із систем не є ідеальною в цьому плані. Кожні із них має як свої плюси, так і мінуси, при чому в деяких випадках недоліків більше, ніж переваг. Для виконання проекту дипломної роботи було проаналізовано кожен із недоліків, для уникнення повторення таких же недоліків у своїй роботі, а задля забезпечення швидкодії системи, обрано тільки необхідні переваги.

3 ВИБІР ПІДХОДІВ ТА ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ

3.1 Технологічний стек

Під час розробки проєкту використовувались різноманітні технології, мови програмування та верстки, а також база даних. Технологічний стек даного проєкту складається з:

- мова гіпертекстової розмітки HTML
- каскадна таблиця стилів CSS
- мова програмування JS
- мова програмування PHP
- база даних MySQL
- SQL – запити для зв'язку проєкту із базою даних

Опишемо детальніше кожен із елементів технологічного стеку.

3.2 Мова гіпертекстової розмітки HTML

HTML – мова гіпертекстової розмітки, яка призначена для того, щоб відкривати сторінки, написані у текстовому редакторі за допомогою веб-браузера. Якщо документ збережено у форматі .html, то у ньому повинні бути:

1. Прописані рядки, які визначають, файл саме html файлом;
2. заголовок в тезі HEAD;
3. тіло документа, поміщене в тег BODY.

Мова HTML до HTML5 версії визначалась як додаток SGML. Специфікації HTML5 формулюються в об'єктній моделі документа, більш відомій, як DOM.

HTML документ складається з певного набору елементів, де кожен елемент визначається певним тегом. Тег – це назва елемента, записана в кутових дужках (<>).[7]

З кожною версією перелік тегів змінювався, оскільки технології постійно розвиваються, а разом з ними розвиваються і браузери. Тому в різні періоди технологічного розвитку то з'являлась необхідність певного тегу, то пропадала,

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		23

оскільки його можна було замінити іншим, більш ефективним тегом або технологією.

Структуру HTML документу візуально можна представити так:

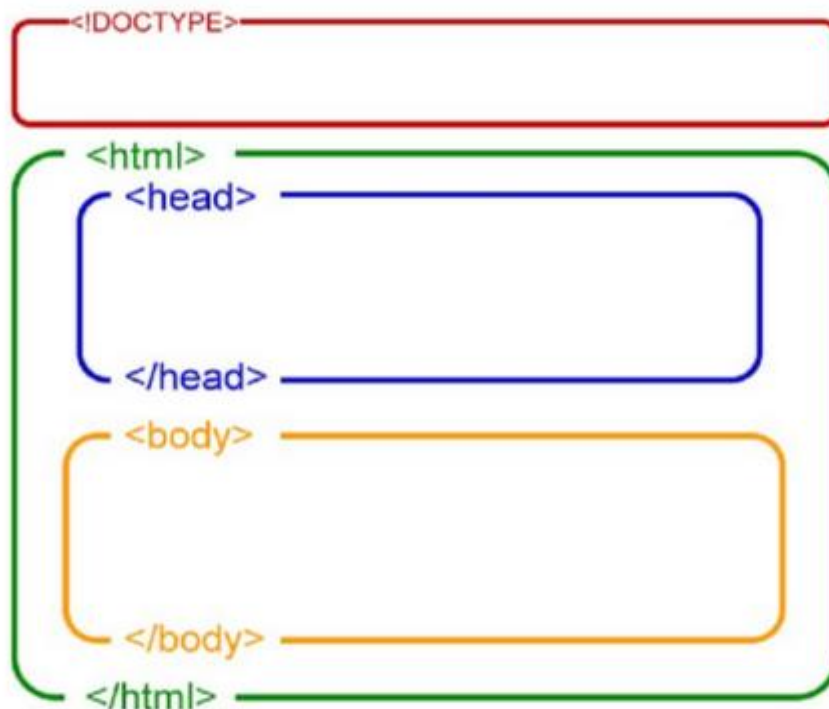


Рисунок 3.1 – Структура HTML документу [8]

- `<html>`, `<head>`, `<body>`
- `<div>`, `<form>`
- `<table>`, `<tr>`, `<td>`, `<th>`
- `<h1>`, `<p>`, ``, ``, `<i>`, `<pre>`
- `<input/>`, `<textarea>`
- `<audio>`, `<video>`, ``

Рисунок 3.2 – Перелік основних HTML тегів

3.3 Каскадна таблиця стилів CSS

CSS – стильова мова, завдання якої полягає в тому, щоб надати сторінці у форматі html стильового оформлення. Його робота полягає в тому, щоб дати змогу розробникові розмежувати вміст веб-сторінки від його оформлення. CSS працює за таким механізмом – текст спочатку виводиться, а потім підлаштовується під властивості, які описані в таблиці стилів. При хорошому інтернет-з’єднанні даний процес переходу від нестилізованого до стилізованого тексту є практично незамітним. Проте його можна з легкістю побачити, коли з’єднання з мережею повільне. [9]

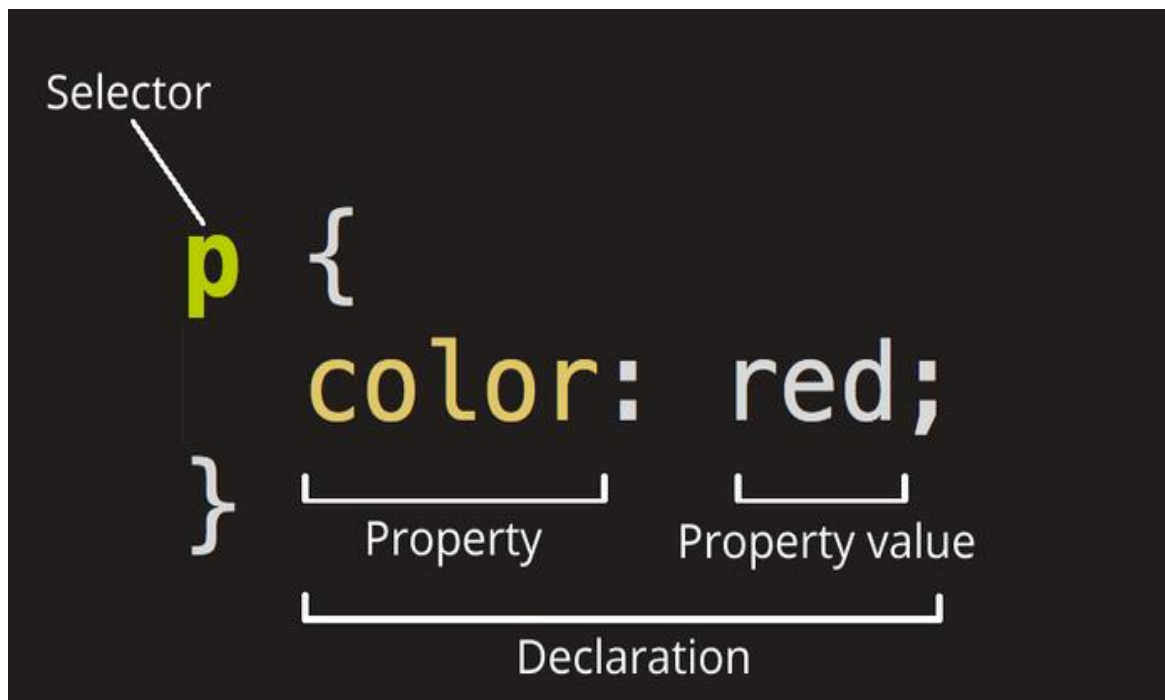


Рисунок 3.3 – Структура CSS правила

Як видно з рисунку 3.3, структура CSS правила складається з селектору, властивості та значення. Пара властивість – значення називається оголошенням.

Селектор – назва HTML елемента, до якого буде застосовуватись оголошення.

Властивість – засоби, якими можна стилізувати HTML документ. Це може бути колір, шрифт тексту, розмір тексту, тощо.

Значення – величина, яка може бути вибрана з-поміж багатьох доступних величин, які можуть застосовуватись до даної властивості.

Сукупність оголошень для певного селектору утворюють правило.

Набір таких правил для невеликого веб-сайту може сягати сотень рядків коду, тому для оптимізації роботи та створення зручних умов для розробника було створено препроцесори. Найвідомішими препроцесорами є:

- SASS/SCSS
- LESS
- Stylus
- PostCSS

Найчастіше застосовують препроцесори SASS/SCSS та LESS.

SASS – це розширення CSS, яке надає можливість використовувати вкладені правила, міксини, імпорти та інше. Він дозволяє зберігати таблиці стилів добре організованими, а невеликим стилям працювати швидко.

LESS – це надбудова над CSS. LESS додає багато динамічних властивостей в CSS. Він вводить змінні, операції, елементи функцій.

3.4 Мова програмування JS

JavaScript – це об'єктно-орієнтована мова програмування, яка найчастіше використовується для створення сценаріїв веб-сторінок. Це надає можливість вести асинхронний обмін даними між клієнтською частиною застосунку, та серверною. Окрім того, ця мова програмування дозволяє змінювати зовнішній вигляд сторінки, змінювати її структуру та керувати браузером. [6]

JavaScript класифікують як прототипну мову програмування з динамічною типізацією.

Мова JavaScript застосовується для:

- надання інтерактивності сторінкам, та написання сценаріїв подій
- створення односторінкових та прогресивних вебзастосунків
- програмування на стороні сервера

- стаціонарних застосунків
- мобільних застосунків
- сценаріїв в прикладних програмах

Деякі веб-сторінки вимагають, щоб в налаштуваннях браузера було надано дозвіл на використання JavaScript, оскільки він їм потрібен для правильної роботи. На рисунку 3.5 зображено один із багатьох можливих варіантів виведення тексту “Hello World!” на веб-сторінці.

```

<!DOCTYPE HTML>
<html>
<body>
  <h1></h1>
  <script type="text/javascript">
    var myHeading = document.querySelector('h1');
    myHeading.textContent = 'Hello world!';
  </script>
</body>
</html>

```

Рисунок 3.4 – Виведення тексту “Hello World!”

3.5 Мова програмування PHP

PHP – популярна серед веб-розробників мова програмування, який найчастіше використовують як серверний інструмент для генерації HTML коду, який пізніше інтерпретується браузером.

У теперішній час виділяють три основні сфери застосування мови PHP:

1. Для програмування скриптів та повноцінних застосунків, які виконуються на стороні сервера. Це одна із найпопулярніших сфер застосування, оскільки спочатку мова створювалась саме для застосування у веб-сайтах на серверній стороні. Для правильної роботи веб-застосунку, написаного на мові PHP, необхідно мати сервер, парсер і клієнтське програмне забезпечення (веб-браузер), які відображають результат роботи коду.

2. Для написання сценаріїв, які виконуються у командному рядку. Такі програми призначені для рооти на будь-якому ПК. Для їх роботи потрібно тільки парсер. В більшості випадків, такі програми створюють для роботи із текстовими даними, так як PHP має потужні інструменти для роботи із текстом.

3. Для створення графічного інтерфейсу. PHP має велику кількість засобів, призначених для реалізації різних завдань. Одним із таких інструментів є PHP-GTK. Як правило, його використовують програмісти, які мають досвід роботи із синтаксисом PHP. [10]

Перевагами цієї мови програмування є:

- простий та зрозумілий синтаксис
- кросплатформність та гнучкість
- масштабованість
- вбудованість в HTML-документи
- детальна документація
- широкий спектр застосування
- популярність на світовому ринку
- наявність великої кількості готових рішень

PHP дозволяє легко поєднати клієнтську частину веб-сайту із базою даних, оскільки SQL запити, які необхідні для зв'язку із таблицею бази даних, можна прописати прямо в ній, а потім отриману інформацію вивести прямо в частину HTML коду.

На рисунку 3.5 зображено схему роботи веб-застосунку із серверною частиною на мові програмування PHP.

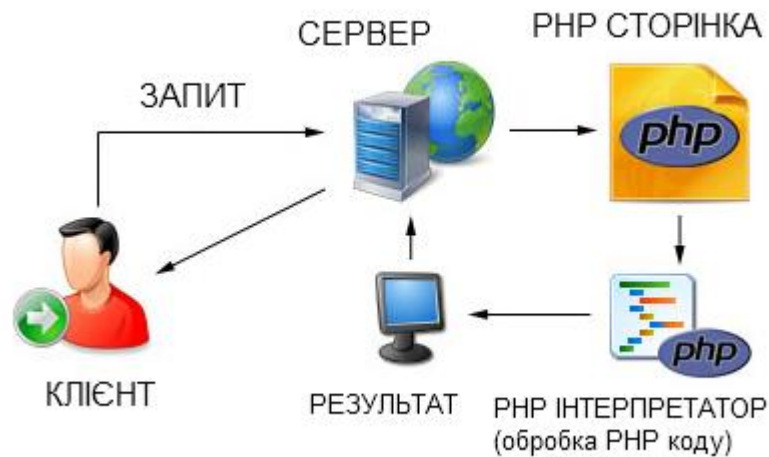


Рисунок 3.5 – Схема роботи веб-сайту на основі PHP [11]

На рисунку 3.6 наведено приклад виведення тексту “Hello World!” прямо в HTML коді веб-сторінки.

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title> Hello World Program </title>
5    </head>
6
7    <body>
8
9      <?php
10       echo "Hello World!";
11     ?>
12
13   </body>
14
15 </html>
  
```

Рисунок 3.6 – Виведення тексту “Hello World!”

Для того, щоб PHP код виконувався прямо у веб-сторінці, як це показано на рисунку 3.6, документ повинен мати розширення .php і запускатись з використанням веб-серверу, або його емулятором.

3.6 База даних MySQL

MySQL – це система керування реляційними базами даних. Вона використовує модель «клієнт - сервер».

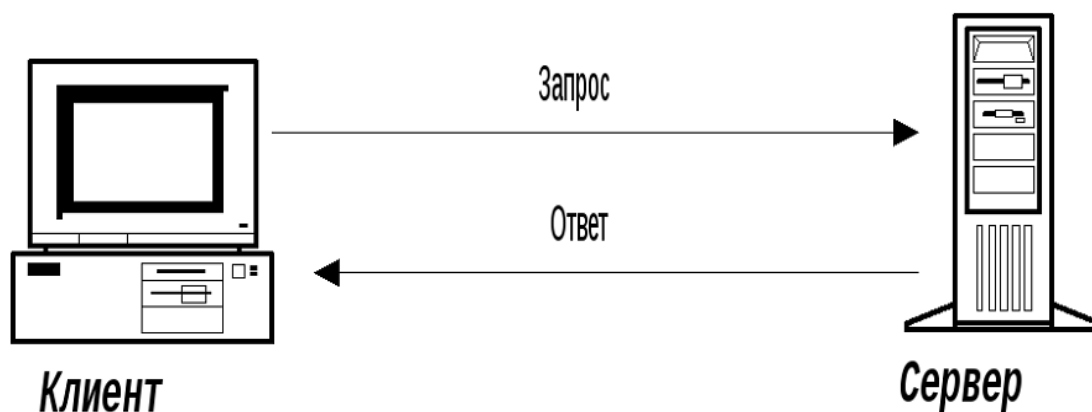


Рисунок 3.7 – Модель роботи БД MySQL

Великі застосунки, такі як Google, YouTube, Twitter, Facebook та Yahoo! використовують MySQL як засіб для збереження власних даних.

Перевагами MySQL є:

- простота – є можливість зміни коду самостійно
- продуктивність – система підтримує велику кількість моделей кластерних серверів
- стандартизація – підтримується промисловий стандарт, а замовники досить швидко отримують замовлене програмне забезпечення
- безпека – наявна система доступу до облікових записів, яка дозволяє зберегти цілісність та безпеку бази даних. Крім того доступне шифрування пароллю і перевірка на основі хосту. [12]

Базова структура системи клієнт-сервер дуже проста. Клієнт підключається до сервера через мережу. Він запитує інтерфейс користувача. Якщо сервер зрозумілий інструкції, він видає інформацію.

У MySQL відбуваються такі процеси:

- створення бази даних MySQL;

- запити користувачів, використовуючи команди SQL;
- відповідь сервера інформацією, яка надсилається користувачу.

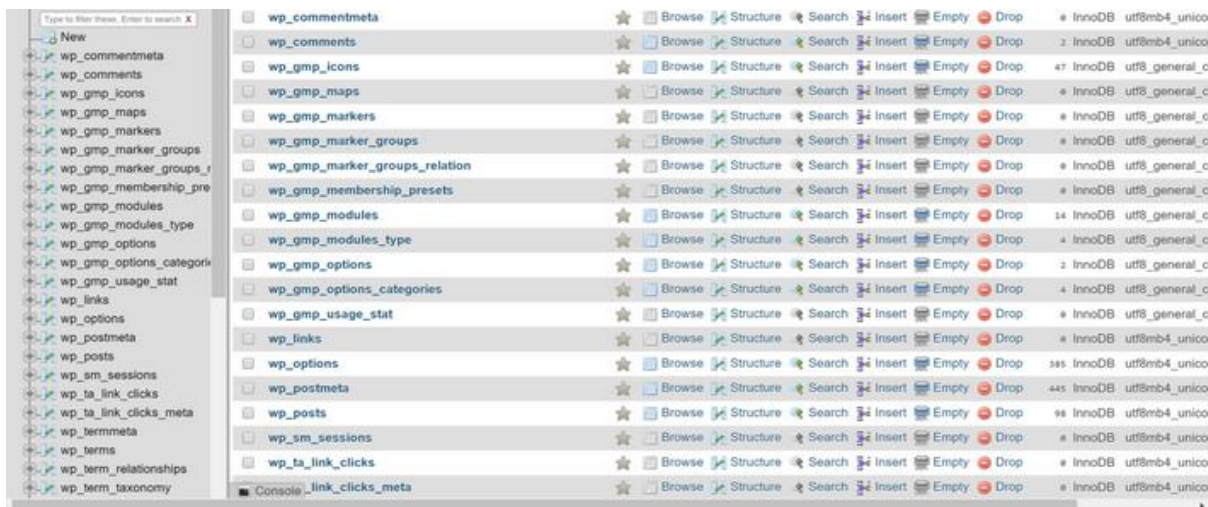


Рисунок 3.8 – База даних на основі MySQL

3.7 SQL-запити

SQL – мова запитів до сховища даних. Вона використовується для зв'язку серверної частини застосунку та клієнтської. Під час його виконання серверу необхідно виконати наступні дії:

- отримання даних із певного сховища інформації;
- редагування даних (додавання, сортування, видалення тощо);
- ідентифікацію отриманих даних;
- забезпечення секретності даних та їх безпеки.

3.7 Середовище розробки Visual Studio Code

Visual Studio Code – це професійний редактор коду, який може бути використаний для написання програм різними мовами програмування. Розробникам надається потужний програмний засіб для розробки з великим функціоналом та приємним інтерфейсом. Користувач може відкрити один або декілька каталогів проектів для одночасної роботи над кількома проектами з одного

ПК. У ньому присутня підтримка великої кількості мов програмування та надбудов над стандартними опціями продукту.

Окрім того, VS Code має велику кількість плагінів, які можна завантажити. Це полегшує розробку застосунку, оскільки плагіни дозволяють пришвидшити написання проєкту шляхом скорочення написання основних елементів застосунку. Також, плагіни дозволяють стилізувати візуально робочу область, тому написання коду в цьому програмному засобі приносить тільки задоволення.

Висновок до розділу: у даному розділі було надано список необхідних технологій, які повинні бути застосовані у даному проєкті для задоволення тих цілей, які були поставлені під час планування проєкту. Було проведено короткий опис кожної із технологій.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		32

4 РОЗРОБКА ЗАСТОСУНКУ

4.1 Архітектура сервісу

Веб-застосунок — це застосунок, розроблений на основі клієнт-серверної моделі, де браузер виконує роль клієнтської сторони, а веб-сервер - серверної. Робота веб-застосунку поділяється на клієнтську та серверну. Дані зберігаються на серверній стороні або у файлах куки. Обмін даних мережею виконується за допомогою використання протоколу передачі гіпертексту HTTP.

Перевагою такого підходу є те, що користувач не залежить від операційної системи, в якій запускається веб-застосунок. Таким чином, веб-додатки є крос-платформними сервісами.

Розробка клієнтської частини — це процес створення інтерфейсу для користувача, за допомогою якого він буде взаємодіяти зі всіма механізмами системи. Структура веб-сторінки, заснована на обробленнях подій, таких як натискання клавіші клавіатури, або клік миші.

Розробка серверної частини застосунку стосується реалізації сторони, яка фокусується на логіці веб-додатків. Це створення функціоналу веб-застосунку, розробка платформи. Серверна сторона управляє даними, отриманими з клієнтської частини, і повертає результат користувачеві у формі, яка зрозуміла в першу чергу йому. Серверна частина сайту складається з: веб-сервера, логіки програмного застосунку та бази даних.

Програмне забезпечення веб-сервера — це частина застосунку, яка працює на стороні клієнта та призначена для відображення даних користувачеві у веб-браузері. Програмне забезпечення веб-серверу складається з кількох частин, а його ядром є HTTP сервер. Логіка його роботи така: зі сторони браузера надсилається запит, за допомогою написаного програмного забезпечення дані подаються на сервер, після цього сервер аналізує інформацію, та повертає відповідь на отриману інформацію.

При використанні протоколу HTTP завжди виконується певний набір правил. По-перше, сервер здатний відповідати тільки на запит, який був надісланий

					IA93.110BAK.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		33

користувачем. По-друге, сервер надсилає відповідь на кожне повідомлення, отримане з клієнтської сторони. Це може бути навіть повідомлення про помилку на одному із етапів оброблення повідомлення. По-третє, кожен запит повинен містити посилання на сервер та шлях доступу, куди потрібно надіслати відповідь. Логіка програми в більшості випадків називається бізнес-логікою сервера. Вона вміщає в собі операції, які використовуються для обробки запитаних та надісланих даних, надсилання даних до сховища даних для подальшого їх зберігання, отримання інформації про необхідні дані, а потім, їх запит з бази даних. Її застосування дозволяє додавати до застосунку допоміжні засоби безпеки, такі як механізми аутентифікації для визначення користувача, який надсилає дані до сервера та отримує їх назад.

Ще одним із заходів безпеки веб-застосунку є авторизація. Вона застосовується для того, щоб перевірити, чи присутній користувач у системі, та чи має він права на виконання певних функцій у сайті.

Опираючись на вищезазначену інформацію, можна зробити висновок, що бізнес-логіка є однією із найважливіших частин роботи, оскільки вона надає можливість створювати, зберігати та перенаправляти дані з різних частин застосунку. На рисунку 4.1 зображено важливість використання бізнес-логіки.

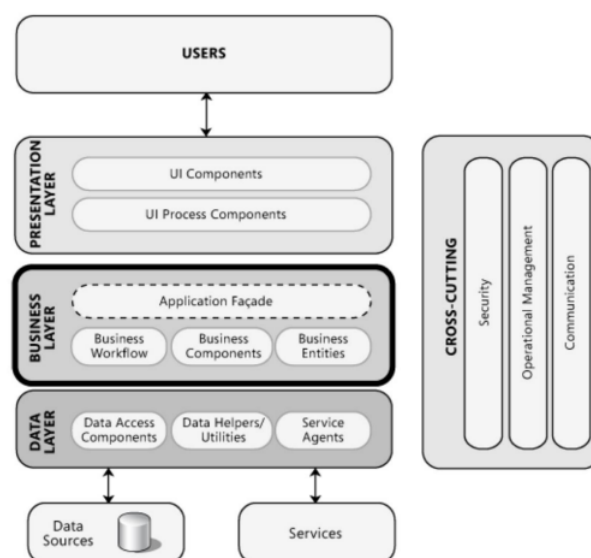


Рисунок 4.1 — Структура веб-застосунку

З рисунку 4.1 видно, що бізнес-логіка займає одну з найважливіших частин у структурі програми. Саме у ній реалізуються функції, за допомогою яких пізніше будуть опрацьовуватись дані, надіслані користувачем, та повертатись дані, які користувач очікує побачити.

Система управління базами даних (СУБД) відіграє надзвичайно важливу роль у розробці веб-застосунків, тому що використовується для зворотнього зв'язку із клієнтською стороною додатку. СУБД надає можливості для збереження, модифікації та видалення даних. За її допомогою можна отримати інформацію з конкретної бази даних для передачі інформації на сторону клієнта, наприклад, для виведення вмісту таблиці для наведення переліку футбольних матчів. Бази даних допомагають контролювати перелік користувачів системи, список транзакцій, перелік подій та інші дані у форматі, коли звичайний користувач ніяким чином не зможе отримати доступ до них.

База даних MySQL славиться тим, що це ПЗ з відкритим кодом, тому велику кількість подій, які відбуватимуться у БД можна автоматизувати. MySQL відноситься до реляційних баз даних, а це значить, що всі дані зберігаються у формі таблиць. Таблиці можуть зв'язуватись між собою за допомогою відносин, що забезпечує можливість отримання даних з кількох таблиць одночасно.

Система безпеки бази даних заснована на паролях з можливістю верифікації з віддаленого комп'ютера. При передачі паролів мережею, останні шифруються, що забезпечує надійність бази даних та унеможливлення доступу до БД сторонніх осіб.

Через вищезазначені переваги, для даного дипломного проекту було прийнято рішення використовувати реляційну БД MongoDB. Всі запити до бази даних реалізуються за допомогою скриптової мови програмування PHP, оскільки вона забезпечує легкість отримання даних із клієнтської сторони застосунку, та відправлення їх до БД.

4.2 Розробка веб-застосунку

					ІА93.110БАК.005 ПЗ	Арк.
						35
Зм.	Лист	№ докум.	Підпис	Дата		

Розроблена система — це веб-застосунок для замовлень квитків на футбольні матчі, який розроблений для економії часу на очікуванні квитків за рахунок модернізації процесу отримання квитка та власне квитка. Для розробки клієнтської частини застосунку було обрано мову гіпертекстової розмітки HTML в парі з каскадною таблицею стилів CSS та мовою програмування JS для забезпечення динамічності оброблення даних у веб-застосунку. Для роботи із серверною частиною програми було обрано мову програмування PHP. Розроблена система забезпечує більшу частину функцій, яку повинна містити аналогічна система, проте за бажанням, до системи з легкістю можна додати більше функціональних можливостей.

У розроблюваній системі було розроблено наступні інтерфейси для роботи:

- для реєстрації або авторизації користувача в системі;
- для перегляду квитків;
- для перегляду доступних місць;
- для бронювання місця;
- для перегляду замовлень;
- для редагування переліку матчів (інтерфейс адміністратора).

Як видно з попереднього переліку, веб-застосунок складається із двох інтерфейсів; користувача та адміністратора.

4.3.1 Модуль системи керування переліком матчів

У цьому модулі зібрано операції, за допомогою яких адміністратор має змогу керувати переліком матчів та властивостями кожного із існуючих матчів.

Додавання матчу. Додавання матчу в перелік існуючих матчів є основною функцією панелі адміністратора. Інформація про матчі, що вводиться тут, буде відображатися на сторінці веб-сайту у розділі «Квитки» або на головній сторінці веб-застосунку.

Коли адміністратор хоче додати новий матч, він повинен натиснути відповідну кнопку в панелі адміністратора. Перед ним відкриється форма, в яку він

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		36

повинен внести необхідні дані. Після цього, за допомогою SQL скриптів, записаних в обробнику форми, дані будуть внесені до списку матчів в базі даних. Після проведення цієї процедури, у списку матчів на стороні користувача можна буде побачити щойно доданий матч у відповідному розділі.

Ще одна з основних функцій адміністратора — редагування матчу. Для того, щоб адміністратор мав змогу відредагувати інформацію про певний матч, на панелі адміністратора він повинен натиснути кнопку «Редагувати» у рядку того матчу, який повинен бути змінений. Після цього адміністратор має ввести актуальні дані про матч, та зберегти зміни. Після проведення цієї процедури, користувач матиме змогу бачити перед собою змінену інформацію про певний матч.

Окрім того, для адміністрування сайту, додано можливість видалити певний матч. Для цього адміністратор просто повинен натиснути кнопку «Видалити» у тому рядку, де розміщений матч, який необхідно прибрати зі списку. Ця функція необхідна тоді, коли матч уже розпочався і продаж квитків на нього припинено, або коли матч не відбувається з якихось причин.

Для взаємодії клієнтської частини з сервером використовується PHP. У даному застосунку він працює як обробник подій, оскільки він дозволяє зібрати дані з клієнтської частини та надіслати їх на сервер для подальшого зберігання. Окрім того, він застосовується у клієнтській частині застосунку для відображення вмісту таблиці бази даних.

Для оброблення даних використовується мова програмування JavaScript, оскільки вона має більше інструментів для роботи з ними.

Якщо отримані дані із сервера збігаються із тими, що уже змодельовані, то на сторінці не відбувається нічого. Якщо дані, отримані із сервера, відрізняються від уже змодельованих, то сторінка змінює свій стан так, щоб відобразити актуальні дані сервера.

4.3.2 Модуль онлайн замовлень

Відображення інформації про матчі на загальнодоступній сторінці веб-сайту. Інформація про матчі, додані в системі керування, повинна бути відображена на

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		37

загальнодоступній сторінці веб-сайту, на яку користувач може перейти в браузері за посиланням .../event/:id.

На головній сторінці та в розділі меню «Квитки» користувач має змогу переглянути доступні матчі, які додаються у відповідну категорію, до якої їх відніс адміністратор. Матчі додаються по черзі у кожену категорію відповідно.

Замовлені клієнтами квитки, будуть зберігатися в кошику. Реалізація алгоритмів збереження даних у кошику досягається за рахунок використання сесій PHP. [13]

Сесії надають можливість збереження невеликої кількості даних, які:

- не вносяться до бази даних;
- при повторному відвідуванні сторінки користувачем, не будуть збережені в системі.

Сесія створює власний об'єкт на клієнтській стороні, у який можна зберігати довільна дані у форматі пар «ключ - значення». Для того, щоб дані почали зберігатись в сесії, потрібно відобразити, які дані повинні бути збережені в них. Для цього необхідно прописати `session_start()`, після цього помістити дані, які потрібно зберегти.

Коли користувач знову відкриє вікно браузера, перевірка підкаже, що у даного користувача уже є створений файл сесії, і відобразить інформацію, яку було збережено. В нашому випадку, користувач отримає квиток у розділі «Кошик». У випадку редагування змін, у файл сесій запишеться нова інформація, яка буде відображатись наступні рази.

Авторизація. Компонент авторизації дозволяє користувачу увійти у власний обліковий запис використовуючи тільки його електронну пошту та пароль.

Для реалізації авторизації у PHP-обробнику відбувається перевірка даних, введених користувачем, і даних, які відповідають введений електронній пошті в базі даних. У випадку, коли дані збігаються, відбувається вхід користувача у систему, а у файл-кукі вводиться поточна електронна пошта користувача. У випадку, коли дані не збігаються, користувач отримує помилку входу і повинен ввести пароль ще раз.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		38

Окрім того, в обробнику PHP відбувається перевірка, на наявність аккаунту в базі даних під час процесу реєстрації. Це необхідно для запобігнення запису двох аккаунтів під однією електронною скринькою, яка уже існує у базі даних, і введені дані збігаються із тими, що у БД, то користувача буде записано у файл-кукі і буде виконано вхід в систему. Проте, коли дані відрізняються, користувачеві буде виведено помилку про невідповідність введених даних і буде запропоновано вхід ще раз.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/> 1	id	int(100)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
<input type="checkbox"/> 2	name	text	utf8_general_ci		Ні	Немає			Змінити Знищити Більше
<input type="checkbox"/> 3	surname	text	utf8_general_ci		Ні	Немає			Змінити Знищити Більше
<input type="checkbox"/> 4	email	varchar(50)	utf8_general_ci		Ні	Немає			Змінити Знищити Більше
<input type="checkbox"/> 5	password	varchar(50)	utf8_general_ci		Ні	Немає			Змінити Знищити Більше

Рисунок 4.2— Структура таблиці users

Як видно з рисунку 4.3.3, для реєстрації користувача, потрібно ввести 4 поля, оскільки поле id служить ідентифікатором і додається в базу даних автоматично. Для авторизації необхідно правильно ввести тільки 2 поля: електронна пошта та пароль.

4.3.3 Модуль обробки замовлення

Після того, як користувач оплатив квитки, які він обрав як бажані до покупки, замовлення потрапляє до бази даних в таблицю orders. Інформація з цієї таблиці пізніше використовується для генерування QR коду, оскільки в ній поміщено інформацію про місце та матч, які обрав користувач.

Наявність усіх даних в одній таблиці значно спрощує генерацію QR-коду, оскільки вибірка елементів відбувається одним SQL-запитом, а це мінімізує витрачання трафіку мережі та пришвидшує генерацію квитка, оскільки він генерується тільки після завантаженні сторінки.

Використання такого підходу зменшує час на очікування відкриття сторінки, цим же виконує один із пунктів завдання веб-застосунку – веб-сайт повинен швидко відкривати різні сторінки додатку.

Проте у такого підходу є незначний недолік – при нестабільному підключенні до мережі генерація QR-коду може зайняти декілька секунд. Протягом цього часу на сторінці не виводиться повністю квадрат QR-коду, але і відсутнє повідомлення про те, що кошик пустий.

Для генерації QR-коду використовується сервіс api.qrserver.com. [14] Він один із небагатьох сервісів, який надає можливість використання програмного API, що дозволяє динамічно створювати QR-коди у власних проектах.



Рисунок 4.3 – Згенерований QR-код у проекті

Якщо відсканувати QR-код, зображений на рисунку 4.3, можна побачить наступну інформацію: ID матчу, на який було придбано квиток та перелік місць, які були обрані користувачем. Наявність такої інформації унеможливить генерації однакового QR-коду, оскільки ID може бути тільки унікальним, а база даних вносить інформацію про заброньовані місця і унеможлиблює подальший вибір цього місця.

Простота використання такого принципу виділяється тим, що для перевірки такого квитка не потрібно виготовляти спеціальні пристрої для зчитування інформації – для перевірки квитка достатньо відсканувати QR-код звичайним сканером, який вбудовується у більшість сучасних смартфонів. Це означає, що

використання пристроїв для потрапляння на стадіон зводиться до мінімуму – потрібно два смартфони. Один – для замовлення квитка, другий – для перевірки QR-коду.

4.4 Діаграма послідовності

Діаграма послідовності – це UML-діаграма, у якій лінії життя застосунку відображаються у формі упорядкованої послідовності дій. У графічному матеріалі представлена діаграма послідовності ІА93.110БАК.005 Д1, у якій графічно описано порядок оформлення замовлення та інші можливі сценарії роботи застосунку відповідно до функціональних вимог веб-застосунку.

В процесі замовлення квитка користувач обирає місця, які він бажає додати до покупки. Після обрання всіх бажаних місць користувач повинен натиснути кнопку «Придбати».

Якщо на момент натиску на кнопку «Придбати» кількість місць в кошику менше одного, застосунок забороняє оформити замовлення і в розділі «Кошик» буде виводитись інформація, що заброньовані місця відсутні.

Якщо користувач обрав хоча б одне місце та підтвердив вибір, користувачеві виводиться форма, де він вводить персональні дані і здійснює оплату. Після вводу користувачем інформації відбувається їх перевірка на коректність: ПІБ має бути не пустим, e-mail і номер телефону повинні відповідати стандартному формату.

Якщо хоча б одне із введених полів не відповідає нормам, користувач отримує помилку проведення покупки з можливістю виправлення інформації. Якщо ж усі дані введено правильно, інформація надсилається на сервер, а з серверної сторони надходить повідомлення про успішне проведення замовлення.

					ІА93.110БАК.005 ПЗ	Арк.
						41
Зм.	Лист	№ докум.	Підпис	Дата		

4.5 Діаграма розгортання

У графічному матеріалі наведено діаграму розгортання ІА93.110БАК.005 Д2. Програма складається із трьох частин: клієнтської частини, серверної частини та сервера бази даних .

Ієрархія їх роботи наступна: сервер сховища даних, у який входить компонент MongoDB; Веб-сервер, який складається із двох компонентів: презентаційний рівень та інтерфейс бази даних; Робоча станція, у яку входить Веб-браузер.

4.6 Діаграма діяльності

У графічному матеріалі наведено діаграму діяльності ІА93.110БАК.005 Д3. Початком діяльності системи є запуск застосунку на пристрої користувача. Після запуску застосунку користувачеві стає видимою початкова сторінка. Після натискання на кнопку «Квитки», користувача переадресовує на сторінку з каталогом матчів, де він може обрати бажаний. Після натискання на будь який матч, користувач переходить на сторінку, де може додати у кошик бажане місце на трибуні. Система може перейти в кінцевий стан, тобто зупинити свою роботу, в будь який момент діяльності, якщо користувач закрив додаток.

4.7 Діаграма компонентів

У графічному матеріалі наведено діаграму компонентів ІА93.110БАК.005 Д4. У даній діаграмі наведено процес замовлення квитка та його перегляд на веб-сайті проекту.

Першим етапом є перегляд місць, доступних до покупки. Потім іде етап обрання місця та, за умови, що користувач виконав вхід в обліковий запис, придбання квитка. Кінцевим етапом є перегляд згенерованого квитка в розділі «Корзина».

Висновки до розділу: у даному розділі було описано архітектуру веб-застосунку, проведено аналіз механізму роботи застосунків на основі архітектури

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		42

«клієнт-сервер». Крім того, було описано механізм роботи розробленого застосунку, обґрунтовано вибір технічної складової застосунку, таких як: обрання мов програмування для клієнтської та серверної сторони додатку, обрання бази даних. Також було описано методику зберігання частини даних без додавання їх у базу даних – за допомогою сесій.

Було описано інформацію, зображену на кресленнях до роботи.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		43

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Мета тестування

Метою тестування є перевірка достовірності роботи веб-застосунку для замовлення квитків на футбольні матчі, та окремих його функцій.

5.2 Загальні положення

Як і будь-який програмний продукт, перед представленням застосунку його слід протестувати, щоб визначити певні недоліки у роботі програми, або її слабкі місця. Розрізняють ручне, автоматизоване тестування та напівавтоматизоване тестування.

Тестування програмного забезпечення - це процес технічного дослідження, який дозволяє оцінити відповідність програмного забезпечення на відповідність вимогам та контексту, в якому буде полягати його подальша робота. Крім того, тестування дозволяє виявити помилки в роботі застосунку, які не було виявлено на етапі розробки веб-сайту. Поділ тестувань відбувається на статичне, що проводить перевірку застосунку на відповідність специфікації та технічному завданню, та динамічне, що проводиться в результаті запуску застосунку та запису у відповідні поля різноманітних даних з метою знаходження невідповідності у роботі або збоїв. Розроблюваний додаток було протестовано за допомогою функціонального та інтеграційного тестування.

5.3 Функціональне тестування

Функціональне тестування – це тестування програмного продукту на відповідність системи функціональним вимогам. Це значить, що під час такого тестування проводиться перевірка системи на уміння вирішувати певні задачі, які повинні опрацьовуватись після певних дій користувача. Функціональні вимоги визначають, що саме робить програмне забезпечення, які завдання воно вирішує.

					ІА93.110БАК.005 ПЗ	Арк.
						44
Зм.	Лист	№ докум.	Підпис	Дата		

Розглянемо основні функції системи та перевіримо їх відповідність вимогам технічного завдання.

Тест № 1. Правильне відображення головної сторінки на ПК.

Вхідні дані:

Ціль: Перевірити правильність відображення стартової сторінки. У верхній частині сторінки повинна відображатися кнопка з посилання на вхід в обліковий запис, посередині сторінки повинна відображатися кнопка з посиланням на сторінку зі списком матчів та сторінку новин.

Результат: тест пройдено.

Тест № 2. Коректне відображення квитків.

Вхідні дані: Застосунок відкрито на головній сторінці.

Ціль: Перевірити коректність відображення списку матчів

Хід проведення:

- 1) відкрити головну сторінку;
- 2) перевірити коректність сортування по лігах;
- 3) перейти на сторінку з мапою;
- 4) перевірити відображення місць на трибуні на мапі у вигляді кнопок;
- 5) натиснути на кнопку та переконатися в коректності відображення інформації про подію.

Результат: тест пройдено.

Тест № 3. Коректність вибору місця на трибуні.

Вхідні дані: застосунок відкрито на головній сторінці.

Ціль: Перевірити коректність відображення та додавання інформації в меню в залежності від вибору місць

Хід проведення:

- 1) перейти на сторінку вибору місць на трибуні;
- 2) Натиснути на одне із доступних місць, та переконатись, що воно додалось до переліку бажаних;

					ІА93.110БАК.005 ПЗ	Арк.
						45
Зм.	Лист	№ докум.	Підпис	Дата		

3) натиснути на одне із недоступних місць, та переконатись, що місце не було додано до переліку;

4) перевірити правильність проведення обрахунків під час отримання загальної суми чеку;

5) натиснути на кнопку «Замовити» та переконатись, що сайт надсилає діалогове вікно для проведення оплати квитка.

Результат: тест пройдено.

Тест № 4. Коректність додавання квитка в кошик.

Вхідні дані: Застосунок відкрито на сторінці матчу з переліком місць.

Ціль: Перевірити коректність додавання квитка в кошик товарів.

Хід проведення:

1) натиснути «Замовити» під переліком обраних місць;

2) перейти у розділ «Кошик»;

3) переконатися, що квиток відображається у формі QR-коду;

4) перевірити, чи відповідає інформація про обрані місця дійсності.

Результат: тест пройдено.

Тест № 5. Неможливість оформлення замовлення, якщо не обрано жодного місця на трибуні.

Вхідні дані: Застосунок відкрито на сторінці зі списком місць на трибуні.

Ціль: Перевірити неможливість оформлення замовлення, коли не обрано жодного місця на трибуні.

Хід проведення:

1) переконатися, що для замовлення не обрано жодного місця на трибуні;

2) Натиснути кнопку «Замовити»;

3) перейти у розділ «Кошик»;

4) перевірити, що інформація про новий квиток, оформлений користувачем, не була записана.

Результат: тест пройдено.

Тест № 6. Можливість оформлення замовлення, якщо обрано хоча б одне місце на трибуні.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		46

Вхідні дані: Застосунок відкрито на головній сторінці застосунку.

Ціль: Перевірити можливість оформлення замовлення, якщо обрано хоча б одне місце на трибуні.

Хід проведення:

- 1) Обрати бажаний матч;
- 2) Обрати місце на трибуні, доступне для покупки;
- 3) натиснути кнопку «Замовити»;
- 4) Пройти процес оплати;
- 5) Перейти у розділ меню «Корзина»;
- 6) Переконались, що відображається квиток з актуальною інформацією.

Результат: тест пройдено.

Тест № 7. Коректність обмеження інформації для незареєстрованого користувача.

Вхідні дані: Застосунок відкрито на головній сторінці.

Ціль: Перевірити коректність відображення інформації для незареєстрованого користувача.

Хід проведення:

- 1) переглянути пункти в навігаційному меню веб-сайту;
- 2) переконались, що відсутній пункт «Кошика»;
- 3) обрати будь-який доступний матч;
- 4) побачити інформацію, про необхідність реєстрації для оформлення замовлення;

Результат: тест пройдено.

Тест №8. Правильність відображення інформації для зареєстрованого користувача.

Вхідні дані: Застосунок відкритий на головній сторінці, обліковий запис активовано.

Ціль: переконались у правильності відображення інформації для зареєстрованого користувача.

Хід проведення:

					IA93.110BAK.005 ПЗ	Арк.
						47
Зм.	Лист	№ докум.	Підпис	Дата		

- 1) переглянути пункти в навігаційному меню веб-сайту;
- 2) перевірити наявність розділу «Корзина» ;
- 3) перевірити, що після входу в обліковий запис значок «Вхід» змінився на «Вихід з облікового запису»;
- 4) Обрати будь-який матч;
- 5) Обрати будь-яке доступне місце на трибуні;
- 6) Перевірити, що немає інформації про необхідність реєстрації для замовлення квитка.

Результат: тест пройдено.

Тест № 9. Можливість додавання адміністратором нового матчу.

Вхідні дані: Застосунок відкритий на сторінці панелі адміністратора.

Ціль: Переконатися в можливості додавання нового матчу адміністратором.

Хід проведення:

- 1) натиснути кнопку «Додати»;
- 2) ввести необхідну інформацію у форму;
- 3) натиснути кнопку «Додати»;
- 4) переконатися, що матч з'явився в переліку;

Результат: тест пройдено.

Тест № 10. Можливість видалення існуючого матчу адміністратором.

Вхідні дані: Застосунок відкритий на сторінці панелі адміністратора.

Ціль: Переконатися в можливості видалення існуючого матчу адміністратором.

Хід проведення:

- 1) знайти матч, який потрібно видалити зі списку;
- 2) у відповідному рядку натиснути кнопку «Видалити»;
- 3) переконатись, що матч було видалено зі списку;

Результат: тест пройдено.

Тест №11. Можливість редагування існуючого матчу адміністратором.

Вхідні дані: Застосунок відкритий на сторінці панелі адміністратора.

Ціль: Переконатися в можливості редагування існуючого матчу адміністратором.

Хід проведення:

- 1) знайти матч, який потрібно відредагувати;
- 2) у відповідному рядку натиснути кнопку «Редагувати»;
- 3) заповнити форму для редагування актуальною інформацією;
- 4) натиснути кнопку «Підтвердити»;
- 5) переконатись в тому, що інформація про матч оновилась.

Результат: тест пройдено.

Тест № 12. Можливість реєстрації користувача в системі

Вхідні дані: Застосунок відкрито на головній сторінці.

Ціль: Перевірити можливість реєстрації нового користувача в системі.

Хід проведення:

- 1) натиснути на іконку «вхід»;
- 2) обрати форму для реєстрації;
- 3) ввести дані;
- 4) натиснути кнопку «Реєстрація»;
- 5) переконатися, що було виконано вхід користувача в систему;

Результат: тест пройдено.

Тест №13. Можливість авторизації користувача в системі

Вхідні дані: Застосунок відкрито на головній сторінці.

Ціль: Перевірити можливість авторизації користувача в системі.

Хід проведення:

- 1) натиснути на іконку «вхід»;
- 2) обрати форму для авторизації;
- 3) ввести дані;
- 4) натиснути кнопку «Вхід»;
- 5) переконатися, що було виконано вхід користувача в систему;

Результат: тест пройдено.

Тест №14. Правильність відображення інформації в розділі «Корзина»

					ІА93.110БАК.005 ПЗ	Арк.
						49
Зм.	Лист	№ докум.	Підпис	Дата		

Вхідні дані: Застосунок відкрито на головній сторінці, вхід в обліковий запис користувача виконано.

Ціль: Перевірити правильність відображення інформації в розділі «Кошик».

Хід проведення:

- 1) обрати будь-який матч;
- 2) обрати будь-яке місце на трибуні;
- 3) провести оплату квитка;
- 4) перейти у розділ «Кошик»;
- 5) переконатися, що в кошику знаходиться інформація про обране місце та квиток у вигляді QR-коду;
- 6) перевірити, чи відповідають вказані місця тим, що було обрано у п.2;
- 7) відсканувати QR-код;
- 8) впевнитись, що надані дані про матч відповідають дійсності.

Результат: тест пройдено.

Тест №15. Перевірка збереження інформації після виходу з браузера

Вхідні дані: Застосунок відкрито в розділі «Корзина», квиток на матч придбано.

Ціль: перевірити правильність збереження інформації в файлах сесій та файлах-кукі.

Хід проведення:

- 1) відсканувати QR-код;
- 2) запам'ятати інформацію, описану в QR-коді;
- 3) закрити браузер;
- 4) відкрити головну сторінку веб-застосунку в браузері;
- 5) пройти авторизацію на сайті;
- 6) відкрити розділ «Корзина»
- 7) перевірити наявність інформації про обране місце та QR-коду в корзині;
- 8) відсканувати QR-код;
- 9) порівняти отримані дані з даними, отриманими в п.1.

Результат: тест пройдено.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		50

5.4 Інтеграційне тестування

Інтеграційне тестування - повна перевірка програмного продукту після його збірки з метою виявлення помилок, що виникають в процесі інтеграції програмних модулів або компонентів. Інтеграційне тестування виконується після модульного тестування та перед валідацією.

Тестування проводились шляхом інтеграційного та функціонального тестування. В результаті проведення цього етапу роботи було перевірено роботу функцій системи та їх відповідність технічному завданню. Для кожного проведеного тестування було сформовано звіт тестування, у якому вказано:

- початковий стан системи;
- вхідні дані;
- хід проведення тестування;
- результат тесту.

Після проведення першого тестування було виправлено незначні недоліки в роботі системи та проведено тестування застосунку ще раз. В кінцевому результаті, всі тестування було пройдено успішно. Крім того, було проведено інтеграційне тестування, в якому перевірялась правильність відображення інформації на пристроях з різною шириною екрану. В результаті можна сказати, що вся інформація була виведена правильно на кожному з пристроїв.

Тест №1. Робота із веб-застосунком на пристрої з шириною екрану 1440px і більше.

Вхідні дані: користувач відкриває застосунок на пристрої із шириною екрану більше 1440px.

Ціль: перевірити коректність представлення інформації на сторінці.

Результат: сторінка відображається правильно.

Тест №2. Робота із веб-застосунком на пристрої з шириною екрану 1024px і більше.

					ІА93.110БАК.005 ПЗ	Арк.
						51
Зм.	Лист	№ докум.	Підпис	Дата		

Вхідні дані: користувач відкриває застосунок на пристрої із шириною екрану більше 1024px.

Ціль: перевірити коректність представлення інформації на сторінці.

Результат: сторінка відображається правильно.

Тест №3. Робота із веб-застосунком на пристрої з шириною екрану 768px і більше.

Вхідні дані: користувач відкриває застосунок на пристрої із шириною екрану більше 768px.

Ціль: перевірити коректність представлення інформації на сторінці.

Результат: сторінка відображається правильно.

Тест №4. Робота із веб-застосунком на пристрої з шириною екрану 425px і більше.

Вхідні дані: користувач відкриває застосунок на пристрої із шириною екрану більше 425px.

Ціль: перевірити коректність представлення інформації на сторінці.

Результат: сторінка відображається правильно.

Тест №6. Робота із веб-застосунком на пристрої з шириною екрану 375px і більше.

Вхідні дані: користувач відкриває застосунок на пристрої із шириною екрану більше 375px.

Ціль: перевірити коректність представлення інформації на сторінці.

Результат: сторінка відображається правильно.

Тест №7. Робота із веб-застосунком на пристрої з шириною екрану 320px і більше.

Вхідні дані: користувач відкриває застосунок на пристрої із шириною екрану більше 320px.

Ціль: перевірити коректність представлення інформації на сторінці.

Результат: сторінка відображається правильно.

На рисунку 5.1 зображено інтерфейс системи на пристрої з шириною екрану

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		52

менше 768px. На рисунку 5.2 зображено інтерфейс системи на пристрої з шириною екрану більше 1024px.



Рисунок 5.1 – Відображення застосунку на смартфоні

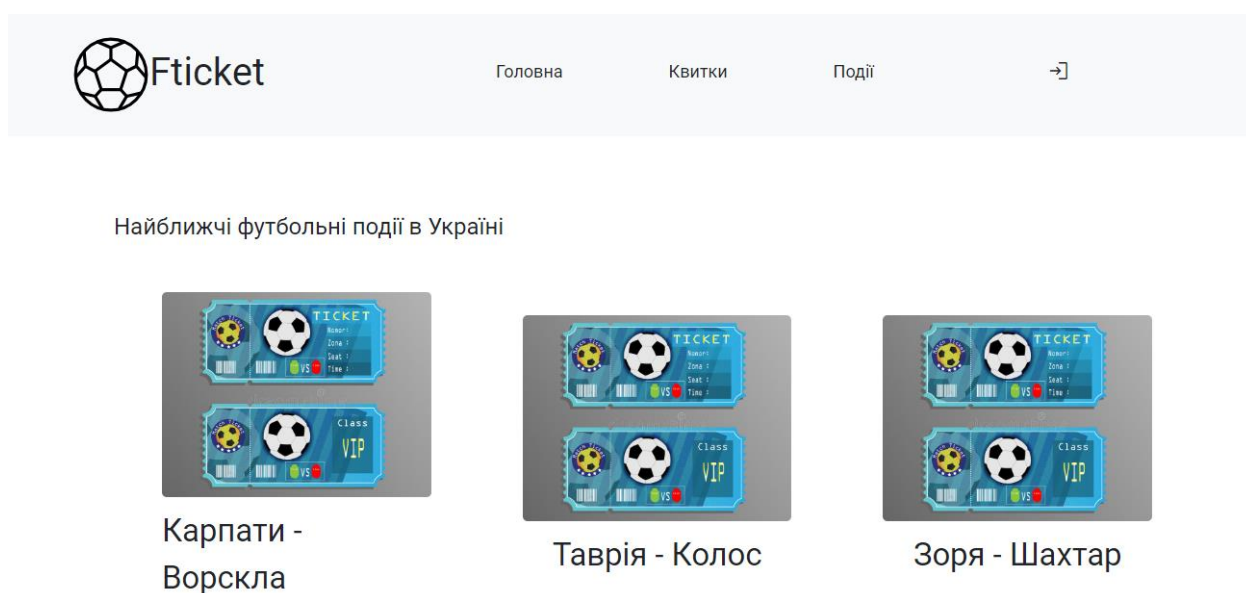


Рисунок 5.2 – Відображення застосунку на ПК

Зм.	Лист	№ докум.	Підпис	Дата

Висновки до розділу: в даному розділі було охарактеризовано рівні тестування, описано необхідність проведення тестування у проєктах, описано механізм роботи функціонального та інтеграційного тестування. Було проведено 15 тестів функціонального тестування та інтеграційне тестування.

В результаті перевірки було доведено, що система працює належним чином, всі тести були пройдені. Інтеграційне тестування довело, що система підтримує адаптивний дизайн та правильно відображає інформацію як на великих екранах ПК, так і на малих екранах смартфонів.

Перевірка відображення застосунку проводилась на наступних розширеннях пристроїв (їх брейкпоінтами): 320px (Mobile S), 375px (Mobile M), 425px (Mobile L), 768px (Tablet), 1024px (Laptop) і 1440px (Laptop L). Помилки не було виявлено, всі елементи на всіх розширеннях екрану відобразилися коректно.

6 ПОРІВНЯННЯ ПРОЄКТУ ТА ГОТОВИХ РІШЕНЬ

На рисунку 6.1 зображено головну сторінку застосунку, розробленого в процесі виконання дипломної роботи.

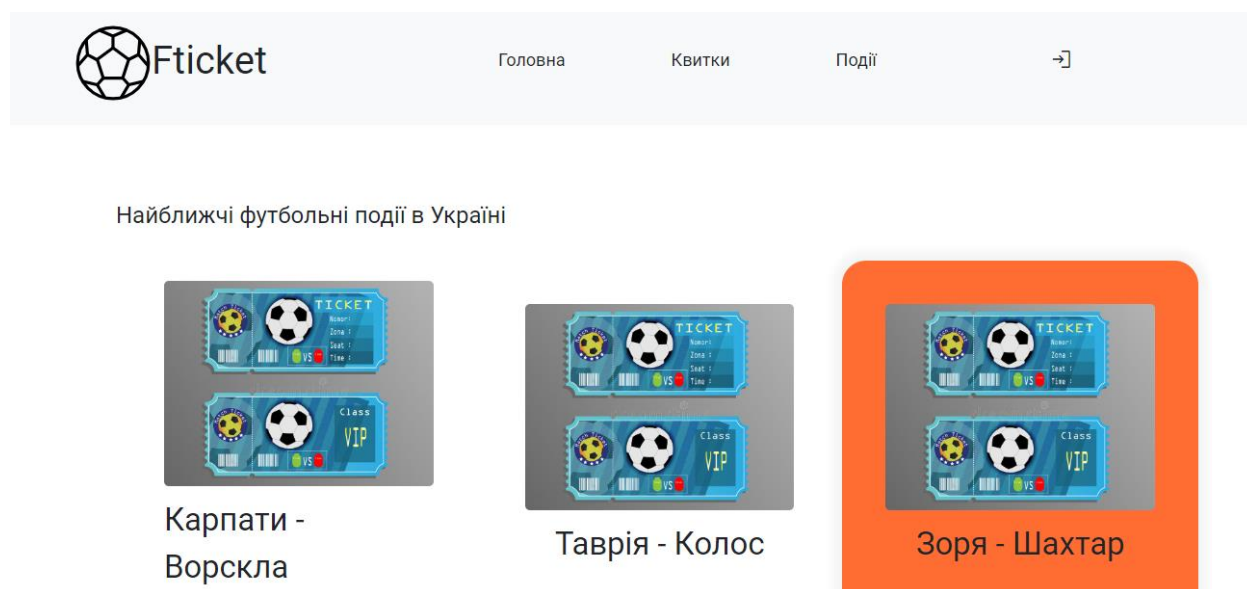


Рисунок 6.1 – Головна сторінка проекту

Як видно з рисунку 6.1, сторінка виділяється одночасно простотою і сучасністю дизайну. Для кращого візуального ефекту, обраний матч підсвічується. На сайті відсутня реклама та банери, які заважають користувачеві побачити основний зміст застосунку. Все зроблено для того, щоб спростити користування сайтом.

Коли користувач хоче придбати квиток, то доцільно розмістити їх одразу на головній сторінці, щоб час, який витратить на пошук квитка був мінімальним. Такий прийом, окрім розробленого застосунку, використовується на сайті ticketsbox.com.

Для правильного відображення застосунку на мобільних пристроях, застосовано адаптивну верстку. Вона застосовується у всіх інших аналогах, оскільки дозволяє не писати два окремі застосунку для користувачів ПК та смартфона. Проте найбільш наглядно роботу видно в розробленому веб-застосунку, оскільки вигляд на ПК, планшеті та смартфоні відрізняється. Для

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		55

користувачів планшетів було вирішено відображати матчі у дві колонки, оскільки так інформація відображається не занадто великим і не занадто малим розміром шрифту і користувач має змогу одночасно побачити більше матчів.

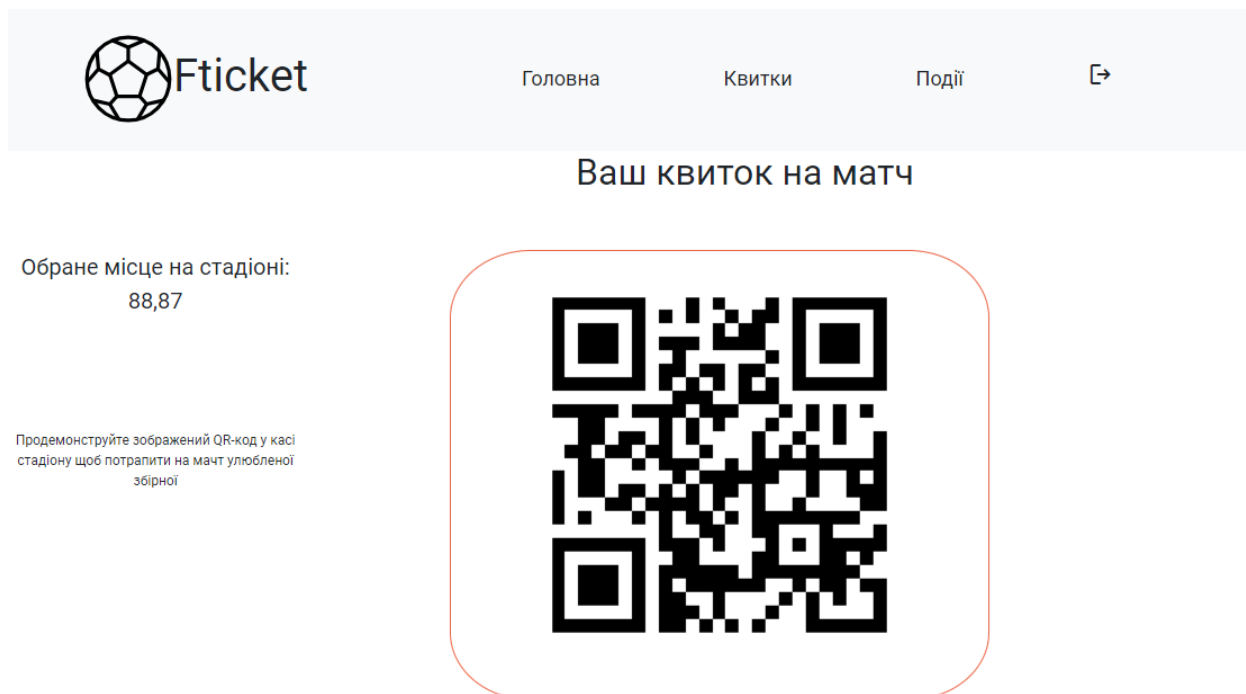


Рисунок 6.2 – Відображення QR-коду в «Корзині»

На рисунку 4.2 зображено QR-код, який був згенерований після покупки квитка на матч. Як видно з рисунку, окрім QR-коду відображається інформація про номер місця на стадіоні, що підтвердить інформацію про те, що QR-код не підроблений. Номер місця на сайті і номер місця після сканування повинен збігатись, тому якщо у них присутні розбіжності, то квиток підроблено.

Така технологія не застосовується у жодному із аналогів, розглянутих у цій дипломній роботі. На даний час, усі сайти, які надають послуги з продажі квитків на футбольні матчі, надсилають квиток з каси на пошту отримувачеві, або взагалі не надають послуги доставки. Натомість, у розробленому застосунку користувач отримує квиток одразу після оплати місця, що значно економить час користувача на отримання квитка. Окрім того, така технологія спрощує процес проходження квиткового контролю, оскільки інформація, яку необхідно перевірити мінімальна.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		56

Висновок до розділу: в даному розділі було описано та зображено на рисунках технології, які роблять розроблену систему рівноправним конкурентом на фоні уже розроблених та активно працюючих систем на ринку України.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		57

ВИСНОВКИ

У результаті написання дипломної роботи було розроблено веб-застосунок для замовлення квитків на футбольні матчі.

У першому розділі – «Аналіз предметної області» було проведено детальний опис предмету розробки, та наведено функціональну модель. За допомогою діаграми варіантів використання було проведено специфікацію функціональної поведінки системи. Було описано, що основними акторами під час роботи із застосунком є три актори: адміністратор – займається створенням нових матчів для відображення їх на сторінці, редагуванням та видаленням існуючих матчів, слідкує за правильністю відображення інформації для користувачів, незареєстрований користувач – актор, який має змогу користуватись обмеженою кількістю функцій веб-додатку, зареєстрований користувач – актор, який має змогу користуватись повним функціоналом застосунку. Сценарії роботи, які доступні користувачам було описано у таблиці.

Відповідно до встановлених варіантів користування системою, було описано функціональні вимоги до програмного застосунку. В процесі розробки було проведено аналіз існуючих рішень систем для замовлення квитків на футбольні матчі. В результаті було обрано три схожі за технологічним стеком та метою сайти, які було проаналізовано, виявлено їх переваги та недоліки, а в останньому розділі, порівняно з розробленою системою.

У розділі програмного та технічного забезпечення було проведено опис технологічного стеку та середовища розробки застосунку. В розділі розробки програмного продукту наведено діаграми: діаграму послідовності, діаграму розгортання, діаграму активності, та діаграму компонентів. Також було проведено аналіз розробки трьох модулів застосунку.

У розділі тестування продукту було сформульовано мету тестування, проведено опис функціонального та інтеграційного тестування та наведено результати кожного із тестів. Випробування проводилися шляхом функціонального та інтеграційного тестування.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		58

Окрім того, було проведено опис основних функцій розробленої системи та перевірено систему на відповідність вимогам технічного завдання. Для кожної вимоги було проведено та описано окреме випробування.

Розроблений застосунок має перспективи подальшого розвитку. При можливому бізнес-рості є сенс розширити функціональну базу приділенням уваги користувацькій частині застосунку.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		59

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mind [Електронний ресурс]. – Режим доступу: <https://mind.ua/news/>
2. TicketsBox [Електронний ресурс]. – Режим доступу: <https://ticketdbox.ua>
3. Skip [Електронний ресурс]. – Режим доступу: <https://esport.in.ua>
4. Kontamarka [Електронний ресурс]. – Режим доступу: <https://kontramarka.ua/uk>
5. Кантор И. Современный учебник Java Script / И. Кантор. – learn.javascript.ru, 2015. – 400 с.
6. JavaScript. TutorialsPoint official website [Електронний ресурс]. – Режим доступу: <https://www.tutorialspoint.com/javascript>.
7. Довідник по HTML тегам [Електронний ресурс]. – Режим доступу: <https://css.in.ua/html/tags>
8. HTML [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/ru/docs/Web/HTML>
9. Основы CSS [Електронний ресурс]. – Режим доступу: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics
10. PHP що це? Чим краще за інші мови програмування? [Електронний ресурс]. – Режим доступу: <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-php/>
11. PHP: MySQL Database [Електронний ресурс]. – Режим доступу: https://www.w3schools.com/php/php_mysql_intro.asp
12. База даних MySQL [Електронний ресурс]. – Режим доступу: <https://promoter.net.ua/articles/baza-danix-mysql.html>
13. Сесії в PHP [Електронний ресурс]. – Режим доступу: <http://www.znannya.org/?view=group:php-sessions>
14. QR code generator [Електронний ресурс]. – Режим доступу: <https://goqr.me/api/>
15. Use case diagram [Електронний ресурс] — Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Use_case_diagram.

16. Deployment diagram [Електронний ресурс] — Режим доступу до ресурсу:
https://en.wikipedia.org/wiki/Deployment_diagram.

17. Component diagram [Електронний ресурс] — Режим доступу до ресурсу:
https://en.wikipedia.org/wiki/Component_diagram.

18. Activity diagram [Електронний ресурс] — Режим доступу до ресурсу:
https://en.wikipedia.org/wiki/Activity_diagram.

19. Я.Ю. Дорогий, К.С. Дорошенко, Н. Б. Репнікова, Л. Ю. Юрчук., Ю.С. Тимофєєва – Дипломний проект бакалавра. Розробка, оформлення, захист[Електронний ресурс]: – Київ : КПІ ім. Ігоря Сікорського, 2022. –78 с.

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		61

ДОДАТОК А

А.1 Лістинг обробника форми реєстрації користувачів

```
<?php
include("db.php");
$name = filter_var(trim($_POST['name']), FILTER_SANITIZE_STRING);
$surname = filter_var(trim($_POST['surname']), FILTER_SANITIZE_STRING);
$email = filter_var(trim($_POST['email']), FILTER_SANITIZE_STRING);
$password = filter_var(trim($_POST['password']), FILTER_SANITIZE_STRING);

$result = $mysql->query("SELECT * FROM `users` WHERE `email` = '$email'");
$user = mysqli_fetch_all($result,1);
if(count($user) != 0){
    $user = $user[0];
    setcookie('user', $user['email'], time() + 3600, "/");
    $mysql->close();
    header('Location: index.php');
    exit();
}

$mysql->query("INSERT INTO `users` (`name`, `surname`, `email`, `password`)
VALUES ('$name', '$surname', '$email', '$password')");

$mysql->close();
header('Location: index.php');
?>
```

А.2 Лістинг коду корзини покупок

```
<!DOCTYPE html>
<html>
```

					ІА93.110БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		62

```

<head>
  <title>Fticket - Квитки на футбол</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFIdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
  <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
  <link rel="stylesheet" type="text/css" href="css/style.css">
  <link rel="stylesheet" type="text/css" href="css/cart.css">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons+Round"
rel="stylesheet">
  <link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght
,FILL,GRAD@20..48,100..700,0..1,-50..200" />
  <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap"
rel="stylesheet">
  <meta name="keywords" content="Fticket, Квитки на футбол">
  <meta charset="utf-8">
</head>
<body onload="generateBarCode()">
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <div class="logo">
      <a href="index.php" draggable="false">Fticket</a>
    </div>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">

```

```

    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
<div class="nav-item links">
    <a href="index.php">Головна</a>
</div>
<div class="nav-item links">
    <a href="tickets.php">Квитки</a>
</div>
<div class="nav-item links">
    <a href="events.php">Події</a>
</div>
<div class="nav-item links">
    <?php
    if(!isset($_COOKIE['user'])){
        ?>
        <a href="signIn.php"><span class="material-symbols-
outlined">login</span></a>
    <?php
    }
    else{
        ?>
        <a href="logout.php"><span class="material-symbols-
outlined">logout</span></a>
    <?php
    }
    ?>
    </div>
</div>
</div>

```

```

</nav>
<?php
    if (!isset($_COOKIE['event'])) {
        echo "Кошик пустий";
    }
    else {
        ?>
<h2 >Ваш квиток на матч</h2>
<div class="qrcode">
    <div class="cartInfo">
        <p style="font-size: 20px;" id="event" hidden="true"><?=$_COOKIE['event']?></p>
        <p style="font-size: 20px;" id="place" hidden="true"><?=$_COOKIE['place']?></p>
        <p style="font-size: 20px;" >Обране місце на стадіоні:
    <?=$_COOKIE['place']?></p>
        <p style="font-size: 12px; margin-top: 100px;">Продемонструйте зображений QR-
код у касі стадіону щоб потрапити на матч улюбленої збірної</p>
    </div>
<?php
    }
    ?>
<div class="img">
    
    </div>
</div>
<script type="text/javascript">
    function generateBarCode()

```

```

    {
        let text = document.getElementById('event').innerHTML;
        let place = document.getElementById('place').innerHTML;
        var url = 'https://api.qrserver.com/v1/create-qr-code/?data=' + "ID of match:"
+ text + " Place : " + place + '&size=50x50';
        $('#barcode').attr('src', url);
        document.getElementById('barcode').style.visibility = 'visible';
    }
</script>

```

```
</body>
```

```
</html>
```

A.3 Лістинг обробника покупки квитка

```
<?php
```

```
$name = $_POST['name'];
```

```
$phone = $_POST['phone'];
```

```
$email = $_POST['email'];
```

```
$tickets = $_POST['tickets'];
```

```
$price = $_POST['price'];
```

```
$event = $_POST['event'];
```

```
include("db.php");
```

```
$list = explode(",",$tickets);
```

```
$count = count($list);
```

```
$mysql->query("INSERT INTO orders(email,places,price,event_id)
```

```
VALUES('$email','$tickets','$price','$event')");
```

```
$mysql->query("UPDATE events SET places=places-$count WHERE id='$event'");
```

```
setcookie('event', $event, time() + 3600);
```

```
setcookie('place', $tickets, time() + 3600);
```

```
header("Location: index.php");
```

```
?>
```

					IA93.110BAK.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		66